Technology Service Corporation

Technology Service Corporation

Washington Operations: 8555 Sixteenth Street, Silver Spring, Maryland 20910 Phone: (301) 565-2970

Ref: TSC-W25-160/rad
B6581111

A REVIEW OF REPRESENTATION FUNCTIONS FOR PROBABILITY OF DETECTION

11 January 1980

by:

Wayne Rivers, J.N. Bucknam,
E.N. Khoury, and R.E. Blase

CDRL Sequence Number A005
Contract N60921-79-C-A306

Prepared for:

Systems Compatibility Branch, Code F-42
Naval Surface Weapons Center
Dahlgren, Virginia 22448

ACKNOWLEDGEMENTS

The work reported here was performed for the Naval Sea Systems Command's Shipboard Surveillance Radar Systems (SSURADS) program under contracts with the Command (N00024-78-C-7065) and the Naval Surface Weapons Center (N60921-79-C-A306). The Command's Program Manager is W. E. French, NAVSEA Code 62X10, and T. Henderson of NSWC/DL Code F-42 represented the Center.

This report is one of a series documenting radar modeling tools for use by Navy Laboratories.

The results reported here are derived almost entirely from the efforts of Jim Bucknam, Ed Khoury, and Bob Blase of TSC. Their memoranda and computer programs are collected here so that a single reference will be available to radar model users.

A REVIEW OF REPRESENTATION FUNCTIONS FOR PROBABILITY OF DETECTION

by

Wayne Rivers, J. N. Bucknam,
E. N. Khoury, and R. E. Blase

ABSTRACT

Simple representation functions that interrelate the primary signal detection variables for the receiver structure of Marcum and Swerling are reviewed in regard to accuracy, complexity and inversion. Functions reviewed include those of Brooks, Neuvy, and Khoury-Bucknam. The first two of these are simple algorithms for computing minimum ratio of signal energy to noise power density as functions of number of samples integrated, the target distribution case, and required probabilities of detection and false alarm. The Khoury-Bucknam functions are analytic and invertable, and they relate probability of detection and signal-to-noise ratio using parameters chosen uniquely for each case, number of samples, and probability of false alarm.

The procedures and software that support determination of the coefficients of the Khoury-Bucknam function are documented.

TABLE OF CONTENTS

1. INTRODUCTION

The process of determining when a desired signal is present in a radar receiver, called detection, was analyzed by Marcum and Swerling [1]* in terms of the receiver structure shown in Figure 1. In the context of Marcum's and Swerling's analysis, the detection process is describable in terms of functions of five (total) variables:

- A target distribution identifier and its associated number of degrees of freedom in the target signal (K);
- The number of signal samples integrated at video (N);
- The probability that the video integrator output exceeds a threshold level, given that no target signal is present: the "probability of false alarm" (P_{FA});
- The probability that the video integrator output exceeds the threshold level, given that the target signal is present: the "probability of detection" (P_D);
- The ratio of energy in a sample of the "target" signal to the background noise power density, the so-called "signal-to-noise ratio" (S).

In addition to the stipulated receiver structure, other parameters of the problem were fixed: the Gaussian distribution of the background noise with known fixed average power (or energy density); ideal and pure target distribution cases belonging to the chi-square family; known time of arrival of the target signal (but not phase); ideal device linearity, or in the case of the envelope "detector" ideal square-law transfer properties; perfect matched filter on the coherent side; large video bandwidth on the video side compared to the matched filter bandwidth;

*Numbers in brackets refer to references in Section 4.

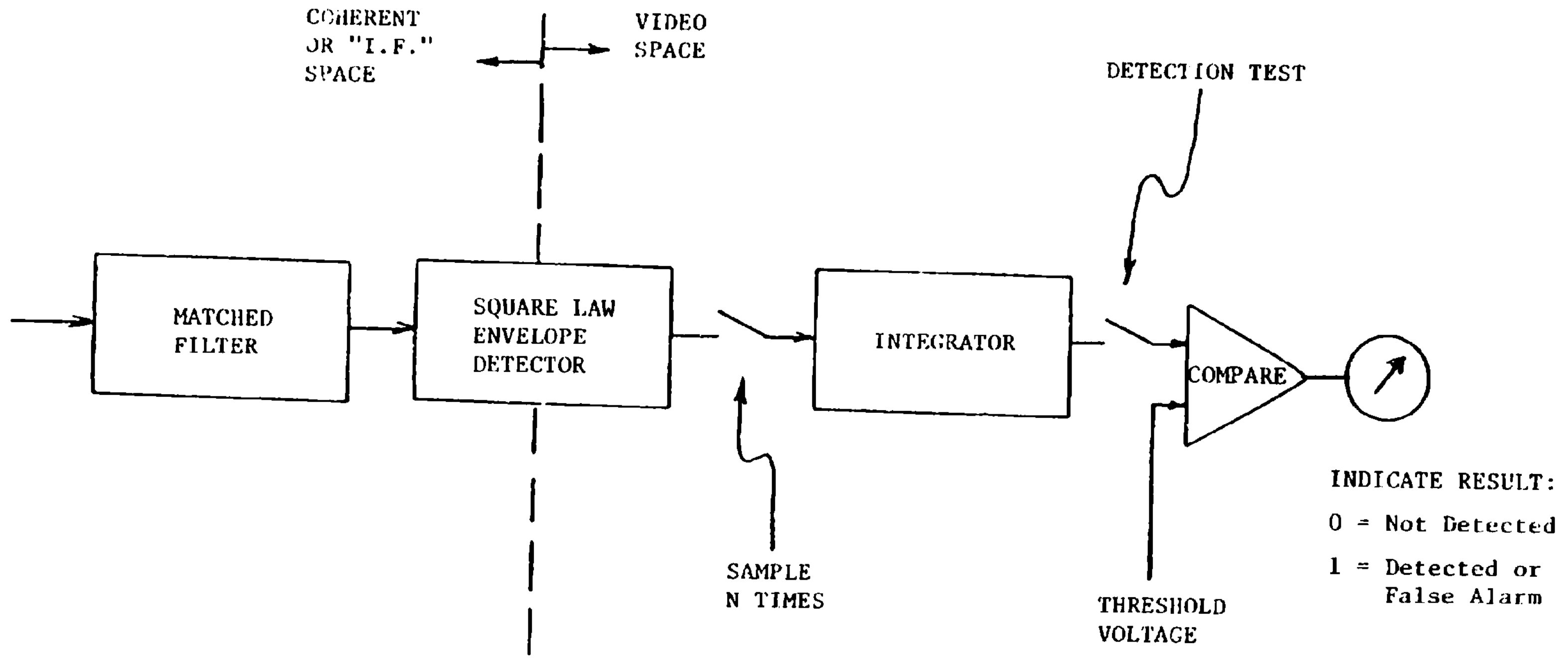

Figure 1 - Receiver Structure for Classical Non-Coherent Radar Detection Analysis

satisfaction of the narrow-band assumption for the coherent matched filter; and perhaps others.

For a given set of radar requirements or an existing radar, appropriate values will be known for the number of samples integrated, the probability of false alarm and the target distribution. Therefore, the last two variables above stand out as principal candidates for independent and dependent status. In modeling radar performance, computation of probability of detection for values of signal-to-noise ratio is a routine and highly repeated activity. The inverse of that function, the signal-to-noise ratio needed to achieve a given probability of detection, is also widely used, but in a less repetitious mode.

Marcum and Swerling [1] published functions of $P_D(S=(R/R_o)^4)$ as plotted curves, with P_{FA}, N, and Case as parameters, and the signal-to-noise power ratio expressed indirectly in terms of a normalized range variable. Meyer and Mayer [2] made computations and presented results as plotted curves of $P_D(N)$, with S_{Db}, $n' = \ln(2)/P_{FA}$, Case as parameters. This latter set of curves is eminently readable and suitable for manual methods of model computation, but the procedure for their computation by machine is lengthy. Fehlner [3] implemented a series-representation of $P_D(S, N, Y_b(P_{FA}, \text{Case}, N), \text{Case})$* which is in use today as part of the NRL model package as subroutine MARSWR [e.g. 4]. Computation cost using this routine can be high (30 to 50% of the total in generating plots of detection performance) and inversion to compute $S(P_D)$ is possible only by iteration. Mitchell and Walker [5] defined a simpler more compact algorithm than MARSWR in terms of recursive properties of the functions that are evaluated by the long series in the other methods. However, its computation costs can be even higher [5].

---

*$Y_b(P_{FA}$, Case, N) is the threshold level in video voltage units.

Over the years, various approximation methods have been defined to reduce the complexity and cost of computation, at the expense of accuracy. North [6] assumed that signal-to-noise ratios of most interest are high so that an asymptotic form of the modified Bessel function could be used, resulting in a marked simplification of the probability integral at a cost in accuracy of only 0.2 dB. Brooks [7] fit the North approximation to produce a simple algorithm amenable to pocket-calculator implementation of the function $S(P_D, N, P_{FA}, \text{Case})$. This algorithm is examined further in the next section. Neuvy [8] under Skolnik's guidance factored the function $S(P_D, N, P_{FA}, \text{Case})$ into the product $S = f_1(P_{FA})\ f_2(N, \text{Case})\ f_3(P_D, \text{Case})$, and fitted the three factors with simple functions. This method is also examined in Section 2.

Khoury and others at TSC recognized the need for a representation both efficient and reasonably accurate for the function $P_D(S)$ which also possessed an inverse $S(P_D)$. Khoury identified such a function having four parameters: A, B, C, and P_{FA}:

$$P_D = 1 - (1 - P_{FA})/(1 + (A \cdot S)^B)^C, \tag{1}$$

and its inverse

$$S = \frac{1}{A}\left(\left(\frac{1-P_{FA}}{1-P_D}\right)^{1/C} - 1\right)^{1/B} \tag{2}$$

Bucknam defined $S_{DB} = 10 \cdot \log_{10}(S)$ and rewrote the functions as

$$P_D = 1 - \left(1-P_{FA}\right)\Big/\left(1 + \left(10^{0.1(S_{DB} + AD)}\right)^B\right)^C \tag{3}$$

$$S_{DB} = \frac{10}{B}\log_{10}\left\{\left(\frac{1-P_{FA}}{1-P_D}\right)^{1/C} - 1\right\} - AD \tag{4}$$

in which $AD = 10 \log_{10}(A)$. Formula 4 is linear in the coefficients AD and 1/B, which fact has significance for the procedure which defines those parameters.

Khoury used sets of pairs of (P_D, S) input to an iterative procedure for finding values of A, B, and C that apply to a particular parameter set (Case, N, P_{FA}) of interest. This procedure is the subject of Sections 3.1 and 3.2 of this report. Buckham derived a procedure taking advantage of the asymptotic properties of Formula 4 and its linear form, and this procedure is the subject of Section 3.3 of this report.

2. COMPARISON OF REPRESENTATION ACCURACIES

The use of approximating representations is appealing because of the much reduced complexity and computation effort. But before approximations can be accepted, users must appreciate the consequences in terms of the accuracy of representation. Here the Brooks, Neuvy, and Khoury-Bucknam (K-B) representations are compared with values read from the Meyer and Mayer curves [2]. Values of S_{DB} are tabulated for all four methods over the following space:

$P_D = 0.01, 0.1, 0.32, 0.5, 0.8, 0.9, 0.99$

$P_{FA} = 10^{-6}, 10^{-8}, 10^{-10}$

$N = 1, 10, 100$

Case = 0, 1, 2, 3, 4

and the results are shown in Tables 1-5. Computation using Brook's and Neuvy's algorithms was straightforward and in accord with the formulas in Appendix A of this report. Coefficients AD, B, and C were obtained using Bucknam's Quick PDFIT recipe applied to the Meyer and Mayer data set as input, and then Formula 4 was used to compute the K-B entries.

The K-B function using Quick PDFIT coefficients departs from the M&M values least for Case 0 (non-fluctuating target) and greatest for Case 1 (Correlated Rayleigh target), with Cases 4, 2, and 3 intermediate. For Cases 0 and 2 ($N \geq 10$) deviations of less than 0.2 dB are typical, whereas for Case 1 0.6 dB differences occur over $0.1 \leq P_D \leq 0.9$ and 1 dB over $0.01 \leq P_D \leq 0.99$. The 0.2 dB error estimate is probably not significant, because the internal consistency of the M&M curves is about 0.2 dB peak-to-peak.

Brook's formulas work well for all cases, with errors generally less than 0.5 dB for $0.1 \leq P_D \leq 0.9$ and less than 1 dB for $0.01 \leq P_D \leq 0.99$.

Neuvy's algorithm fails badly for Case 0, with differences exceeding 3 dB at low values of P_D. For other cases, however, its performance is comparable to but a little poorer overall than Brooks'.

For record purposes, values of the coefficients A, AD, B, and C obtained using Quick PDFIT, and which were used in this comparison, are tabulated in Appendix B of this report.

Table 1 - Comparison of Representation Formulas, Target Distribution Case 0.

P_D	$P_{FA} = 10^{-6}$				$P_{FA} = 10^{-8}$				$F_{FA} = 10^{-10}$			
	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B
N = 1												
.01	+ 5.7	+ 6.4	+ 11.1	+ 5.6	+ 9.0	+ 8.6	+ 12.4	+ 8.9	+ 9.6	+ 10.1	+ 13.4	+ 9.4
.1	8.5	8.8	11.6	8.6	11.0	10.5	12.9	11.1	11.5	11.8	13.9	11.6
.32	10.3	10.5	12.2	10.4	12.3	11.9	13.4	12.4	12.8	13.0	14.4	12.9
.5	11.2	11.2	12.5	11.2	13.0	12.5	13.8	13.0	13.3	13.5	14.7	13.5
.8	12.4	12.4	13.3	12.4	14.1	13.5	14.6	14.0	14.4	14.5	15.6	14.4
.9	13.1	13.0	13.9	13.0	14.5	14.1	15.1	14.5	15.0	15.0	16.1	14.9
.99	14.3	14.3	15.6	14.2	15.6	15.3	16.8	15.6	16.0	16.0	17.8	15.9
N = 10												
.01	- 0.7	- 0.3	.9	- .8	1.7	1.4	2.2	1.6	2.2	2.6	3.1	2.1
.1	+ 1.5	+ 1.6	1.4	+ 1.6	3.3	2.9	2.7	3.5	3.8	3.9	3.6	3.9
.32	3.0	2.8	1.9	3.0	4.5	3.9	3.2	4.5	4.9	4.8	4.1	4.9
.5	3.7	3.4	2.3	3.7	5.1	4.4	3.5	5.1	5.3	5.2	4.5	5.4
.8	4.7	4.3	3.1	4.7	6.0	5.2	4.4	5.9	6.3	6.0	5.3	6.2
.9	5.2	4.8	3.6	5.2	6.4	5.7	4.9	6.3	6.7	6.4	5.9	6.6
.99	6.3	5.8	5.3	6.2	7.4	6.6	6.6	7.4	7.6	7.2	7.6	7.6
N = 100												
.01	- 6.1	- 5.8	- 6.1	- 6.2	- 4.1	- 4.4	- 4.8	- 4.2	- 3.7	- 3.4	- 3.8	- 3.8
.1	- 4.3	- 4.2	- 5.6	- 4.2	- 2.7	- 3.1	- 4.3	- 2.6	- 2.4	- 2.3	- 3.3	- 2.3
.32	- 3.1	- 3.2	- 5.0	- 3.1	- 1.8	- 2.2	- 3.8	- 1.7	- 1.6	- 1.6	- 2.8	- 1.5
.5	- 2.6	- 2.7	- 4.7	- 2.5	- 1.3	- 1.9	- 3.4	- 1.3	- 1.2	- 1.2	- 2.5	- 1.1
.8	- 1.7	- 2.0	- 3.9	- 1.7	- 0.6	- 1.2	- 2.6	- .6	- 0.3	- 0.6	- 1.6	- .5
.9	- 1.2	- 1.5	- 3.3	- 1.3	- 0.2	- 0.8	- 2.1	- .3	- 0.1	- 0.3	- 1.1	- .1
.99	- .3	- 0.7	- 1.6	- .4	+ 0.5	- 0.1	- 0.4	+ .5	+ 0.6	+ 0.4	+ 0.6	+ .6

Table 2 - Comparison of Representation Formulas, Target Distribution Case 1.

P_D	$P_{FA} = 10^{-6}$				$P_{FA} = 10^{-8}$				$P_{FA} = 10^{-10}$			
	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B
N = 1												
.01	+ 3.	+ 2.7	+ 4.7	+ 3.4	+ 4.9	+ 4.2	+ 6.	+ 5.4	+ 6.0	+ 5.4	+ 6.9	+ 6.6
.1	6.9	6.6	7.7	7.5	8.4	8.	9.	9.0	9.4	9.1	9.9	10.
.32	10.3	10.3	10.8	10.4	11.7	11.7	12.	11.6	12.7	12.7	13.	12.6
.5	12.8	12.8	12.9	12.3	14.1	14.1	14.2	13.5	15.	15.1	15.1	14.4
.8	17.9	18.2	17.9	17.1	19.0	19.5	19.1	18.3	20.	20.5	20.1	19.2
.9	21.1	21.8	21.1	20.6	22.4	23.	22.4	21.8	23.3	24.	23.3	22.7
.99	31.4	32.5	31.3	32.2	32.8	33.8	32.6	33.7	33.6	34.7	33.5	34.5
N = 10												
.01	- 4.1	- 4.	- 3.6	- 3.7	- 2.8	- 3.	- 2.3	- 2.3	- 1.8	- 2.1	- 1.3	- 1.1
.1	- 0.5	- 0.7	- 0.5	- .1	+ 0.7	+ 0.6	+ 0.7	+ 1.2	+ 1.5	+ 1.2	+ 1.7	+ 2.1
.32	+ 2.9	+ 2.7	+ 2.5	+ 2.8	4.0	3.7	3.8	3.9	4.8	4.5	4.7	4.6
.5	5.1	5.	4.7	4.7	6.2	6.	5.9	5.7	7.	6.8	6.9	6.4
.8	10.1	10.1	9.6	9.4	11.2	11.2	10.8	10.5	11.9	12.	11.8	11.1
.9	13.5	13.6	12.9	12.9	14.5	14.6	14.1	14.0	15.1	15.4	15.1	14.7
.99	23.6	24.1	23.1	24.5	24.8	25.1	24.3	25.6	25.7	25.9	25.3	26.5
N = 100												
.01	-10.	- 9.6	-10.3	- 9.5	- 8.1	- 8.7	- 9.1	- 7.2	- 8.2	- 8.1	- 8.1	- 7.6
.1	- 6.5	- 6.5	- 7.3	- 6.0	- 5.6	- 5.6	- 6.1	- 4.8	- 5.	- 5.	- 5.1	- 4.4
.32	- 3.3	- 3.3	- 4.3	- 3.4	- 2.4	- 2.5	- 3.	- 2.8	- 1.7	- 1.8	- 2.	- 1.9
.5	- 1.1	- 1.1	- 2.1	- 1.6	- 0.3	- 0.3	- 0.8	- 1.1	+ 0.4	+ 0.4	+ 0.1	- 0.2
.8	+ 3.9	+ 3.9	+ 2.8	+ 3.1	+ 4.7	+ 4.7	+ 4.1	+ 3.7	5.3	5.4	5.	+ 4.6
.9	7.1	7.2	6.1	6.6	8.	8.	7.3	7.3	8.6	8.7	8.3	8.1
.99	17.3	17.5	16.3	18.1	18.1	18.4	17.5	19.3	18.8	19.	18.5	19.7

Table 3 - Comparison of Representation Formulas, Target Distribution Case 2.

P_D	$P_{FA} = 10^{-6}$				$P_{FA} = 10^{-8}$				$P_{FA} = 10^{-10}$			
	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B
N = 1												
.01	+ 3.	+ 2.7	+ 5.1		+ 4.9	+ 4.2	+ 6.4		+ 6.0	+ 5.4	+ 7.3	
.1	6.9	6.6	7.8		8.4	8.	9.		9.4	9.1	10.	
.32	10.3	10.3	10.5		11.7	11.7	11.7		12.7	12.7	12.7	
.5	12.8	12.8	12.4		14.1	14.1	13.6		15.	15.1	14.6	
.8	17.9	18.2	16.7		19.0	19.5	18.		20.	20.5	19.	
.9	21.1	21.6	19.6		22.4	23.	20.9		23.3	24.	21.8	
.99	31.4	32.5	28.6		32.8	33.8	29.9		33.6	34.7	30.8	
N = 10												
.01	- 1.4	- .6	.5	- 1.5	0.2	1.	1.8	+ .04	1.3	2.1	8.7	+ 1.2
.1	+ 1.1	+ 1.4	1.1	+ 1.2	2.3	2.6	2.4	2.5	3.2	3.6	3.3	3.4
.32	2.8	2.8	1.7	2.9	3.8	3.9	3.	3.9	4.6	4.8	4.	4.8
.5	3.7	3.5	2.2	3.7	4.7	4.6	3.4	4.7	5.5	5.4	4.4	5.5
.8	5.3	4.9	3.2	5.2	6.2	5.8	4.4	6.1	7.0	6.6	5.4	6.8
.9	6.2	5.7	3.8	6.0	7.1	6.6	5.1	6.9	7.8	7.3	6.	7.6
.99	8.3	7.6	5.9	8.4	9.1	8.4	7.1	9.3	9.8	9.1	8.1	10.0
N = 100												
.01	- 6.2	- 5.9	- 6.1	- 6.3	- 4.8	- 4.4	- 4.8	-4.9	- 3.8	- 3.4	- 3.8	- 3.9
.1	- 4.3	- 4.2	- 5.6	- 4.2	- 3.3	- 3.1	- 4.3	-3.2	- 2.5	- 2.3	- 3.3	- 2.4
.32	- 3.2	- 3.2	- 5.	- 3.0	- 2.2	- 2.2	- 3.8	-2.2	- 1.6	- 1.6	- 2.8	- 1.5
.5	- 2.6	- 2.7	- 4.7	- 2.5	- 1.8	- 1.8	- 3.4	-1.7	- 1.1	- 1.2	- 2.5	- 1.1
.8	- 1.6	- 1.9	- 3.9	- 1.6	- 0.9	- 1.2	- 2.6	- .9	- 0.3	- 0.6	- 1.6	- .4
.9	- 1.1	- 1.5	- 3.3	- 1.2	- 0.4	- 0.8	- 2.1	- .5	+ 0.1	- .2	- 1.1	- 0
.99	- 0.1	- 0.6	- 1.6	- .2	+ 0.5	+ 0.1	- 0.4	+ .5	1.0	+ .5	+ 0.6	+ 1.

Table 4 - Comparison of Representation Formulas, Target Distribution Case 3.

P_D	$P_{FA} = 10^{-6}$				$P_{FA} = 10^{-8}$				$P_{FA} = 10^{-10}$			
	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B
N = 1												
.01	+ 3.9	+ 4.5	+ 6.2	+ 3.8	+ 6.2	+ 6.4	+ 7.5	+ 6.2	+ 7.2	+ 7.7	+ 8.4	+ 7.2
.1	7.4	7.7	8.2	7.8	9.	9.3	9.5	9.4	10.1	10.5	10.4	10.5
.32	10.2	10.4	10.3	10.3	11.2	11.8	11.5	11.6	12.5	12.9	12.5	12.7
.5	11.8	12.	11.7	11.8	13.2	13.3	13.	12.9	14.2	14.3	13.9	14.0
.8	15.3	15.3	15.	14.8	16.5	16.5	16.2	15.9	17.4	17.5	17.2	16.9
.9	17.2	17.6	17.2	16.8	18.5	18.6	18.4	18.0	19.4	19.5	19.4	18.9
.99	22.8	23.4	24.	23.3	24.1	24.5	25.5	24.8	25.0	25.4	26.4	25.6
N = 10												
.01	- 3.	- 2.1	- 2.	- 3.1	- 1.6	- 0.8	- 0.8	- 1.6	- 0.6	+ 0.2	+ 0.2	- 0.6
.1	+ 0.2	+ 0.5	+ 0	+ 0.5	+ 1.3	+ 1.6	+ 1.2	+ 1.7	+ 2.2	+ 2.5	+ 2.2	+ 2.6
.32	2.7	2.8	2	2.9	3.8	3.8	3.3	3.9	4.6	4.7	4.2	4.7
.5	4.3	4.2	3.4	4.2	5.4	5.2	4.7	5.2	6.1	6.	5.7	6.0
.8	7.6	7.2	6.7	7.2	8.6	8.2	8.	8.1	9.3	9.	8.9	8.8
.9	9.6	9.2	8.9	9.2	10.6	10.1	10.1	10.2	11.3	10.9	11.1	10.9
.99	15.2	14.9	15.7	15.7	16.2	15.8	17.	16.8	16.9	16.5	17.9	17.4
N = 100												
.01	- 8.7	- 7.7	- 8.8	- 8.6	- 7.6	- 6.6	- 7.6	- 7.5	- 6.9	- 5.7	- 6.6	- 6.8
.1	- 6.	- 5.3	- 6.8	- 5.5	- 5.0	- 4.4	- 5.6	- 4.5	- 4.3	- 3.6	- 4.6	- 3.9
.32	- 3.5	- 3.2	- 4.8	- 3.5	- 2.6	- 2.4	- 3.8	- 2.5	- 2.	- 1.7	- 2.5	- 1.9
.5	- 2.	- 1.9	- 3.3	- 2.2	- 1.0	- 1.1	- 2.1	- 1.3	- 0.4	- 0.4	- 1.1	- .7
.8	+ 1.3	+ 1.0	0	+ .7	+ 2.1	+ 1.8	+ 1.2	+ 1.6	+ 2.7	+ 2.4	+ 2.2	+ 2.2
.9	3.2	2.8	2.1	2.8	4.	3.6	3.4	3.6	4.7	4.2	4.3	4.2
.99	8.8	8.4	8.9	9.4	9.6	9.1	10.2	10.2	10.3	9.7	11.2	10.9

Table 5 - Comparison of Representation Formulas, Target Distribution Case 4.

P_D	$P_{FA} = 10^{-6}$				$P_{FA} = 10^{-8}$				$P_{FA} = 10^{-10}$			
	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B	M&M	Brooks	Neuvy	K-B
N = 1												
.01	+ 3.9	+ 4.5	+ 5.8		+ 6.2	+ 6.4	+ 7.1		+ 7.2	+ 7.7	+ 8.1	
.1	7.4	7.7	7.8		9.	9.3	9.		10.1	10.5	10.	
.32	10.2	10.4	9.8		11.2	11.8	11.		12.5	12.9	12.	
.5	11.8	12.	11.1		13.2	13.3	12.4		14.2	14.3	13.4	
.8	15.3	15.3	14.3		16.5	16.5	15.6		17.4	17.5	16.5	
.9	17.2	17.4	16.4		18.5	18.6	17.7		19.4	19.5	18.6	
.99	22.8	23.4	23.		24.1	24.5	24.2		25.0	25.4	25.2	
N = 10												
.01	- 1.1	- 0.4	+ 0.5	- 1.3	+ 0.6	+ 1.2	+ 1.8	+ .5	+ 1.7	+ 2.3	+ 2.8	+ 1.6
.1	+ 1.3	+ 1.5	1.1	+ 1.4	2.6	2.8	2.4	2.7	3.4	3.7	3.3	3.6
.32	2.8	2.8	1.7	2.9	4.	3.9	2.9	4.1	4.7	4.8	3.9	4.8
.5	3.6	3.5	2.1	3.7	4.7	4.5	3.4	4.7	5.3	5.3	4.3	5.4
.8	5.	4.6	3.	5.0	5.9	5.5	4.3	5.9	6.6	6.3	5.3	6.5
.9	5.7	5.2	3.7	5.6	6.6	6.1	4.9	6.5	7.2	6.8	5.9	7.1
.99	7.3	6.7	5.6	7.3	8.2	7.5	6.9	8.2	8.7	8.2	7.8	8.8
N = 100												
.01	- 6.2	- 5.8	- 6.1	- 6.3	- 4.7	- 4.4	- 4.8	- 4.8	- 3.7	- 3.4	- 3.8	- 3.8
.1	- 4.3	- 4.2	- 5.6	- 4.3	- 3.2	- 3.1	- 4.3	- 3.1	- 2.5	- 2.3	- 3.3	- 2.4
.32	- 3.2	- 3.2	- 5.	- 3.1	- 2.2	- 2.2	- 3.8	- 2.1	- 1.6	- 1.6	- 2.8	- 1.5
.5	- 2.6	- 2.7	- 4.7	- 2.5	- 1.7	- 1.9	- 3.4	- 1.7	- 1.2	- 1.2	- 2.5	- 1.1
.8	- 1.6	- 1.9	- 3.9	- 1.7	- 0.9	- 1.2	- 2.6	- .9	- 0.4	- .6	- 1.6	- .5
.9	- 1.2	- 1.5	- 3.3	- 1.2	- 0.5	- .8	- 2.1	- .6	0	- .2	- 1.1	- .1
.99	- 0.2	- .7	- 1.6	- .2	+ 0.3	0	- .4	+ .3	+ 0.8	+ .5	+ .6	+ .8

3. DETERMINATION OF COEFFICIENTS OF THE KHOURY-BUCKNAM FORMULAS

Here are discussed the three principal tools currently in use for finding values of the coefficients A, B, and C, (also denoted as P1, P2, and P3) in Formulas 1-4 of Section 1.

3.1 Program PDFIT

PDFIT calculates a set of values of S corresponding to a set of 12 values of P_D using the recursive procedure of Mitchell and Walker [5] in an iterated successive approximation mode. Using trial values of the three coefficients and Formula (1) of Section 1, the mean square error in P_D is computed. Each coefficient in turn is perturbed by a small amount to compute partial derivatives and the matrix procedure outlined in Khoury's memorandum in Appendix C of this report is used to compute new values of the coefficients. This procedure is iterated until either a time limit is exceeded or the coefficient values have settled. The input data set (S, P_D), the computed values of P_D using Formula 1, and the error are printed to the operator along with the values of the coefficients. Note that PDFIT yields coefficients that minimize the mean square error in $P_D(S)$.

Figure 3-1A shows the dialogue between the operator and the program PDFIT in a typical example, and Figure 3-1B shows the formatted output that resulted. The program PDFIT along with its subroutines, including implementations of the matrix coefficient adjusting procedure and the Mitchell-Walker procedure are given in Appendix D of this report.

```
OK, PDFIT
GO


      PROGRAM PDFIT COMPUTES AN EMPIRICAL FIT TO THE
   EXACT PD VS. SNR CURVE DESIRED. THE EXACT CURVE IS
   SPECIFIED BY SWERLING CASE NUMBER, PROBABLITY OF
   FALSE ALARM, AND NUMBER OF PULSES NON-COHERENTLY
   INTEGRATED. THIS PROGRAM WILL RETURN THE THREE
   PARAMETERS WHICH GIVE THE BEST FIT IN THE MMSE
   SENSE (CF. REF. TSC-W21-40)
      PDFIT WAS DEVELOPED BY E. N. KHOURY, AND R. F.
   PUSATERI - JUNE 79.

   INPUT DATA CONSISTS OF THE FOLLOWING:

      A) OUTPUT FILE NAME
      B) SWERLING CASE NUMBER
      C) PROBAALITY OF FALSE ALARM
      D) NUMBER OF CASES TO BE RUN
      E) NUMBER OF PULSES INTEGRATED FOR EACH CASE

   ENTER OUTPUT FILE NAME - 6
OUTW3

   ENTER SWERLING CASE NUMBER - 0 1 2
0

   ENTER PROBABLITY OF FALSE ALARM
   RANGE:  1.E-3 THROUGH 1.E-15
.1E-5

   ENTER NUMBER OF CASES TO BE RUN
   RANGE:  1 THROUGH 100
1

   ENTER NUMBER OF INTEGRATED PULSES FOR EACH CASE
   RANGE:  1 THROUGH 100
10

PROCESSING CASE NUMBER:    1
```

Figure 3-1A - Operator-Program Dialogue in Use of Program PDFIT

CASE NUMBER 1

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
0.0	0.1E-05	10

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
0.3255940	4.1603012	2.5547212

FIT

N	SNR	PD	PDF	ERROR
1	0.1227E 01	0.04918	0.05413	-0.00494
2	0.1445E 01	0.09938	0.10280	-0.00342
3	0.1710E 01	0.19304	[illegible]	0.00028
4	0.1945E 01	0.3022[illegible]	[illegible]	0.00282
5	0.2128E 01	0.39763	[illegible]	0.00290
6	0.2304E 01	0.49248	0.49093	0.00155
7	0.2525E 01	0.60708	0.60814	-0.00106
8	0.2740E 01	0.70655	0.70943	-0.00288
9	0.2973E 01	0.79579	0.79875	-0.00296
10	0.3399E 01	0.90682	0.90608	0.00074
11	0.3749E 01	0.95615	0.95240	0.00375
12	0.4803E 01	0.99722	0.99403	0.00318

MEAN ERROR = -0.0000 RMS ERROR = 0.00285

Figure 3-1B - Content of Output File OUTW3, Resulting from Program PDFIT

3.2 Program QFIT

QFIT operates on arbitrary sets of (P_D, S_{DB}) data input via a previously prepared file. Two options are available for deriving the coefficients of the fit equation. One applies the Quick PDFIT procedure to be described in Section 3.3 and prints a summary of the input data, the errors, and the coefficients. This option yields coefficients that minimize errors in S_{DB} computed using Formula (4) of Section 1. The other option uses the procedure of Program PDFIT operating on the arbitrary input data and produces the same summary as PDFIT. This option yields coefficients that minimize errors in P_D using Formula (1) or (3) of Section 1. For both options, the input data and computed values from the fit formula can be plotted.

Figure 3-2A shows a sample dialogue in which both options and the plot are executed. Figure 3-2B shows the input file for the example and Figure 3-2C the Quick PDFIT output summary. Figure 3-2D shows the output summary for the PDFIT option, and the plot of input data with computed values using both sets of coefficients is shown in Figure 3-2E.

Program QFIT including all of its subroutines is included as Appendix E of this report.

```
OK, R *QFIT
GO
ENTER INPUT FILE NAME,FORMAT 6A2
TEST.WIB
TO USE BOBS VERSION ENTER 1,ELSE 0
1
ENTER OUTPUT FILE NAME, FORMAT 6A2
OUT.WIB1
IF A PLOT OF RESULTS IS DESIRED ENTER 1, ELSE 0
1
ENTER XDIM(IN),YDIM(IN)
7,5
ENTER X AXIS LABEL, FORMAT 32A1
      S/N
ENTER YAXIS LABEL, FORMAT 32A1
      PD
TO DO COMPLETE PDFIT ENIER 1, ELSE 0
1
PDFIT: TERMINATION 3, MAX. ITERATIONS


ENTER OUTPUT FILE NAME - 12
OUT.WIB2
ENTER: CASE PFA NP NB
0,.1E-5,10,4
IF A PLOT OF RESULTS IS DESIRED ENTER 1, ELSE 0
1
ENTER XDIM(IN),YDIM(IN)
7,5
ENTER X AXIS LABEL, FORMAT 32A1

ENTER YAXIS LABEL, FORMAT 32A1

OK, COMO -E
```

Figure 3-2A - Operator-Program Dialogue in Use of Program QFIT

```
6
9.8,.06
10.199,.128
10.599,.276
11,.604
11.398,.909
11.801,.981
```

Figure 3-2B - Content of Input File Test.WIB, Manually-Fed Data

RESULTS OF GFIT

SNDB	SN	P	PFIT	ERR
9.8000	9.5499	0.0600	0.0535	0.0065
10.1990	10.4689	0.1280	0.1355	-0.0075
10.5990	11.4789	0.2760	0.3152	-0.0392
11.0000	12.5892	0.6040	0.6092	-0.0052
11.3980	13.7975	0.9090	0.8777	0.0313
11.8010	15.1391	0.9810	0.9833	-0.0023

FIT PARAMETERS

P=1-(1+(A*SN)**B)**(-C)

A	B	C
0.71290E-01	0.10750E 02	0.34540E 01

Figure 3-2C - Content of Output File OUT.WIB1, Results of Fit to Manually-Fed Data

WIEBULL - .33

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
0.0	0.1E-05	10

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
0.0493103	10.5946312	142.7287598

FIT

N	SNR	PD	PDF	ERROR
1	0.9550E 01	0.06000	0.04773	0.01227
2	0.1047E 02	0.12800	0.12138	0.00662
3	0.1148E 02	0.27600	0.29042	-0.01442
4	0.1259E 02	0.60400	0.59776	0.00624
5	0.1380E 02	0.90900	0.90856	0.00044
6	0.1514E 02	0.98100	0.99817	-0.01717

MEAN ERROR = -0.0010 RMS ERROR = 0.01103

Figure 3-2D - Content of Output File OUT.WIB2, Results of Fit to P_D, SNDB Data from Mitchell & Walker's Routine

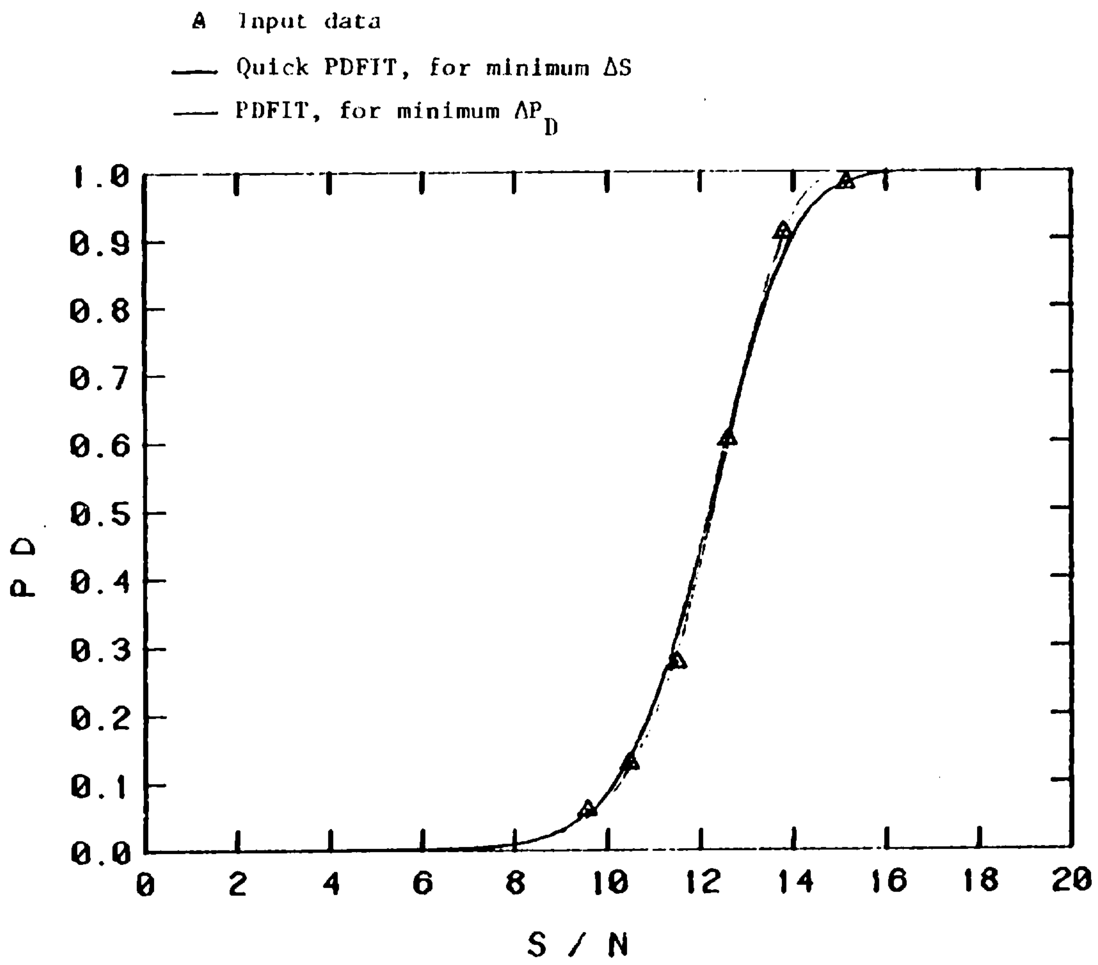

Figure 3-2E - Comparison of PDFIT equations with input data; an output of Program QFIT

3.3 The Quick PDFIT Procedure

Quick PDFIT takes advantage of two properties of Formula 4:

$$S_{DB} = \frac{10}{B} \log_{10} \left\{ \left(\frac{1-P_{FA}}{1-P_D} \right)^{1/C} - 1 \right\} - AD \qquad (4)$$

One is its asymptotic behavior near $P_D \cong 0$ and $P_D \cong 1$; the other is its linear form in coefficients AD and 1/B. The first property is used to obtain an accurate estimate of the coefficient C, by operating on pairs of (P_D, S_{DB}) values only at the extremes. The second property is then used with a simple least-squares procedure operating on all of the (P_D, S_{DB}) data to estimate coefficients AD and B that minimize errors in S_{DB} computed by Formula (4).

Figure 3-3 lists the steps in the Quick PDFIT algorithm, and Figure 3-4 lists the FORTRAN subroutine that implements it. Bucknam's memorandum deriving the procedure is included in Appendix F. Figures 3-3 and 3-4 and Appendix F assume that P_{FA} is either zero or insignificant. For finite P_{FA}, the same procedure can be used by fitting a distorted set of P_D values, given by

$$P_D' = \frac{P_D - P_{FA}}{1 - P_{FA}}$$

Use of the Quick PDFIT procedure on the data pairs (P_D', S_{DB}), with P_{FA} assumed zero, yields values of AD, B, and C for use in the more general expression of equation (4).

No claim is made that the value of coefficient C obtained by the Quick PDFIT procedure is optimum in any sense, although given the initial estimate of C, the other two parameters are optimum for minimum errors in S_{DB}. However, manual adjustment of values of C about the estimated one, followed by recomputation of AD and B, has not been able to show any significant improvement in error over that which results from the Quick procedure. It has been concluded that errors in fitting input data, such as seen in Section 2 of this report, result from intrinsic shape limitations of Formulas (1) - (4) and not from a poor estimate of parameter C.

Figure 3-3 - The Quick PDFIT Algorithm

1. Define function $G(X) = \log_{10}\left\{(1 - X)^{-1/C} - 1\right\}$

2. Input pairs of the variables P_i, S_{DBi}, $i = 1, \ldots n$, in order of increasing values.

3. $Y_i = \log_{10}\left\{-\log_{10}(1 - P_i)\right\}$, $i = 1, 2$

$$M_y = -\frac{Y_2 - Y_1}{S_{DB2} - S_{DB1}}$$

$$Z_i = -\log_{10}(1 - P_i), \quad i = n-1, n$$

$$M_z = \frac{Z_n - Z_{n-1}}{S_{DBn} - S_{DBn-1}}$$

$$C = \frac{M_z}{M_y}$$

4. $Q_i = 1 - (1 - P_i)^{1/C}$, $i = 1, \ldots n$

5. $$M_L = -\frac{G(Q_n) - G(Q_{n-1})}{\log_{10}(Q_n) - \log_{10}(Q_{n-1})}$$

$$M_S = \frac{G(Q_1) - G(Q_2)}{\log_{10}(\frac{1}{C}\ln Q_1) - \log_{10}(\frac{1}{C}\ln Q_2)}$$

$C = C\,M_L/M_S$ (this step adjusts the first estimate of C to an improved value).

Figure 3-3 (continued)

6. $W_i = G(P_i), \quad i = 1, \ldots n$

7. $U_W = \Sigma W_i, \quad i = 1, \ldots n$

$U_S = \Sigma S_{DBi}, \ i = 1, \ldots n$

$U_{SS} = \Sigma (S_{DBi})^2, \ i = 1, \ldots n$

$U_{SW} = \Sigma S_{DBi} \ W_i, \ i = 1, \ldots n$

8. $D = n \cdot U_{SS} - (U_S)^2$

$H_1 = (U_{SS} \ U_W - U_S \ U_{SW})/D$

$H_2 = (N \cdot U_{SW} - U_S \ U_W)/D$

9. $AD = H_1/H_2$

$B = 10 \ H_2$

Figure 3-4 Subroutine QPDFIT

```
      SUBROUTINE QPDFIT (PD,SDB,N,P)
C
C     COMPUTES APPROXIMATE VALUES FOR PARAMETERS
C     P1,P2,P3 IN THE EXPRESSION:
C
C          PD=(1-(1+(P1*S)**P2)**-P3)*100
C
C     WHERE: PD=DETECTION PROBABILITY (IN PERCENT)
C            S=S/N=ALOG10(SDB/10.0)
C
C     SEE TSC-W7-73
C
      DIMENSION Y(2),Z(2),W(200),PD(1),P(3),SDB(1)
      DIMENSION PDD(200)
      REAL MY,MZ
      DO 100 I=1,N
          PDD(I)=PD(I)/100.0
 100  CONTINUE
C
C     COMPUTE SLOPE BETWEEN FIRST TWO POINTS
C
      DO 150 I=1,2
          Y(I)=ALOG10(-ALOG10(1.-PDD(I)))
 150  CONTINUE
      MY=(Y(2)-Y(1))/(SDB(2)-SDB(1))
C
C     COMPUTE SLOPE BETWEEN LAST TWO POINTS
C
      NM1=N-1
      IZ=0
      DO 200 I=NM1,N
          IZ=IZ+1
          Z(IZ)=-ALOG10(1.-PDD(I))
 200  CONTINUE
      MZ=(Z(2)-Z(1))/(SDB(N)-SDB(NM1))
C
C     COMPUTE INITIAL ESTIMATE OF K3
C
      P(3)=MZ/MY
C
C     DISTORT THE PD VALUES
C
      DO 225 I=1,N
          PDD(I)=1.-((1.-PDD(I))**(1./P(3)))
 225  CONTINUE
C
C     COMPUTE LARGE PD SLOPE RATIO
C
      SRL1=ALOG10(((1.-PDD(N))**(-1./P(3)))-1.)
      SRL2=ALOG10(((1.-PDD(NM1))**(-1./P(3)))-1.)
      SRL3=(-1./P(3))*ALOG10(1.-PDD(N))
      SRL4=(1./P(3))*ALOG10(1.-PDD(NM1))
      SRL=(SRL1-SRL2)/(SRL3+SRL4)
C
C     COMPUTE THE SMALL PD SLOPE RATIO
C
      SRS1=ALOG10(((1.-PDD(2))**(-1./P(3)))-1.)
      SRS2=ALOG10(((1.-PDD(1))**(-1./P(3)))-1.)
      SRS3=ALOG10((-ALOG(10.0)/P(3))*(ALOG10(1.-PDD(2))))
      SRS4=ALOG10((-ALOG(10.0)/P(3))*(ALOG10(1.-PDD(1))))
      SRS=(SRS1-SRS2)/(SRS3-SRS4)
      P(3)=P(3)*(SRL/SRS)
C
```

```
C
C     INITIALIZE COUNTERS FOR LINEAR REGRESSION
      SUM=0
      SUMX=0.0
      SUMY=0.0
      SUMX2=0.0
      SUMXY=0.0
C
C     COMPUTE W(I) AND PARAMETERS FOR A STRAIGHT LINE FIT
C
      DO 250 I=1,N
         W(I)=ALOG10(((1.-.01*PD(I))**(-1./P(3)))-1.)
         SUM=SUM+1.0
         SUMX=SUMX+SDB(I)
         SUMY=SUMY+W(I)
         SUMX2=SUMX2+SDB(I)*SDB(I)
         SUMXY=SUMXY+SDB(I)*W(I)
  250 CONTINUE
C
      D=SUM*SUMX2-SUMX*SUMX
      B=(SUMX2*SUMY-SUMX*SUMXY)/D
      A=(SUMXY*SUM-SUMX*SUMY)/D
C
C     COMPUTE K1 AND K2
C
      P(1)=10.**(B/(10.*A))
      P(2)=10.*A
C
      RETURN
      END
```

4. REFERENCES

1. J. I. Marcum, "Studies of Target Detection by a Pulsed Radar-Mathematical Appendix," Rand Research Memorandum RM-753, July 1948.

   P. Swerling, "Probability of Detection for a Fluctuating Target," Rand Research Memorandum RM-1217, March 1954. See also IRE Trans. Vol IT-6, April 1960.

2. D. P. Meyer and H. A. Mayer, Radar Target Detection, Academic Press, 1973.

3. L. F. Fehlner, "Marcum's and Swerling's Data on Target Detection by a Pulsed Radar," Appendix A of Applied Physics Laboratory, The Johns Hopkins University, Report TG-451 (R.G. Rolls and G.T. Trotter, Methods for Digital Machine Computation of Detection Probability), July 1962.

4. L. V. Blake, "A FORTRAN Computer Program to Calculate the Range of a Pulse Radar," NRL Report 7448, August 1972.

5. R. L. Mitchell and J. F. Walker, "Recursive Methods for Computing Detection Probabilities," IEEE Trans., Vol. AES-7, July 1971, p 671-676.

6. D. O. North, "An Analysis of the Factors which Determine Signal/Noise Discrimination in Pulsed Carrier Systems," Proceedings of the IEEE, Vol. 51, July 1963, p 1010-1027.

7. L. W. Brooks, Jr., "Detection Theory on a Pocket Calculator," Proceedings of the IEEE Southeastern Conference, April 1977, p 354-357.

8. J. Neuvy, "An Aspect of Determining the Range of Radar Detection," IEEE Transactions, Vol. AES-6, July 1970, p 514-521.

9. E. N. Khoury, "Curve Fitting P_D vs. SNR Curves," Technology Service Corporation Memorandum TSC-W21-36, March 1979; (Included as Appendix C of this report).

10. J. N. Bucknam, "Quick PDFIT," Technology Service Corporation Memorandum TSC-W7-73, October 1979; (Included as Appendix F of this report).

APPENDIX A

Approximation Algorithms of Brooks and Neuvy

Table A-1 Brooks' Algorithm for Minimum Required Signal-to-Noise Ratio [7]

Reference: L. W. Brooks, Jr., "Detection Theory on a Pocket Calculator," Proceedings of IEEE Southeastcon, April 1977; p 354-357.

Definitions:

P_D = Probability detection occurs, given that target signal is present.

P_{FA} = Probability that detection occurs, given that target signal is not present → probability of false alarm per cell, per N-sample group.

N = Effective number of samples integrated at video

ξ = Asymptotic efficiency of the envelope detector

K = Number of degrees of freedom in the target signal in N samples

S = Ratio of target signal energy to noise power density that would produce P_D.

Algorithm:

1.

Detector	ξ
Square Law	1.0
Linear Law	0.915
Logarithmic Law	0.608

2.

Target Distribution Case	K
0. Nonfluctuating	∞
1. Rayleigh; Correlated over N	1
2. Rayleigh; Fluctuating samples	N
3. Chi Square, k=2; Correlated over N	2
4. Chi Square, k=2; Fluctuating over N	2N

3. $G_{FA} = 2.36\,[-\log_{10} P_{FA}]^{\frac{1}{2}} - 1.02$

$$T = 1.8\,P_D - 0.9$$

$$G_D = 1.23T/(1 - T^2)^{\frac{1}{2}}$$

$$X = G_{FA} + G_D$$

$$L_F = \left\{-\ln(P_D)\,\frac{X}{G_{FA}}\right\}^{-1/K}$$

$$S = \frac{X^2 L_F}{4N}\,\{1 + [1 + 16N/(\xi X^2)]^{\frac{1}{2}}\}$$

$$S_{DB} = 10\,\log_{10}(S)$$

4. No analytic inverse for Brooks' formula is known.

Table A-2 Neuvy's Algorithm for Minimum Required Signal-to-Noise Ratio

<u>Reference:</u> J. Neuvy, "An Aspect of Determining the Range of Radar Detection," <u>IEEE Trans</u>., <u>AES-9</u>, July 1970; p 514-521.

<u>Definitions:</u>

P_D = Probability detection occurs, given that target signal is present.

P_{FA} = Probability of false alarm, per cell per N-sample group.

N = Effective number of samples integrated at video.

S = Ratio of target signal energy.

<u>Algorithm:</u>

1. $E = \exp\{-N/3\}$

2.

Target Distribution Case	α	β
0. Nonfluctuating	1 + 2E	1/6
1. Rayleigh; Correlated over N Samples	2/3(1+2/3E)	1
2. Rayleigh; Uncorrelated	1	1/6+E
3. Chi-square, k=2; Correlated over N Samples	3/4(1+2/3E)	2/3
4. Chi-square, k=2; Uncorrelated	1	1/6+2/3E

3. $S = \alpha \log_{10}(1/P_{FA})/\{N^{2/3}(\log_{10}(1/P_D))^{\beta}\}$

$S_{DB} = 10 \log_{10}(S)$

4. Inverse: $U = [\frac{\alpha \log_{10}(1/P_{FA})}{S\ N^{2/3}}]^{1/\beta}$

$P_D = 10^{-U}$

APPENDIX B

Coefficients of Khoury-Bucknam Representations

Definitions:

P_D = Probability of Detection

S_{DB} = Minimum Detectable Signal-to-Noise Ratio in dB

N = Number of signal samples integrated at video

P_{FA} = Probability of False Alarm

Case = Target Distribution Case

0 = Nonfluctuating

1 = Rayleigh, Correlated across N samples

2 = Rayleigh, Decorrelated by sample

3 = Chi-square with k=2, Correlated across N samples

4 = Chi-square with k=2, Decorrelated by sample

Formulas:

$$P_D = 1 - (1 - P_{FA})/[1 + (10^{0.1(S_{DB} + AD)})^B]^C$$

$$S_{DB} = \frac{10}{B} \log_{10} \{(\frac{1-P_{FA}}{1-P_D})^{1/C} - 1\} - AD$$

Table B-1 Coefficients of Khoury-Bucknam Representations Used in Comparisons

P_{FA}	N	Coef.	Distribution Case 0	1	2	3	4
10^{-6}	1	A	0.0433	0.1265		0.0912	
		AD	-13.64	-8.98	See	-10.40	See
		B	3.3177	2.6832	Case 1	2.6786	Case 3
		C	4.8476	0.3205		0.5805	
10^{-6}	10	A	0.2938	0.7345	0.3936	0.5309	0.3475
		AD	-5.32	-1.34	-4.05	-2.75	-4.59
		B	4.1934	2.9222	3.7306	2.9538	3.872
		C	3.7524	0.2962	1.2269	0.5250	1.8606
10^{-6}	100	A	1.3366	3.0761	1.349	2.4660	1.3677
		AD	+1.26	+4.88	+1.30	+3.92	+1.36
		B	5.0991	3.0927	4.9281	3.4707	5.0075
		C	3.3728	0.2815	3.0079	0.4316	3.0079
10^{-8}	1	A	0.0380	0.0991		0.0760	
		AD	-14.20	-10.04	See	-11.19	See
		B	4.5679	3.0893	Case 1	3.2739	Case 3
		C	2.7339	0.2742		0.4504	
10^{-8}	10	A	0.263	0.5861	0.3373	0.4395	0.3090
		AD	-5.80	-2.32	-4.72	-3.57	-5.10
		B	5.5040	3.0900	4.2905	3.2360	4.6284
		C	2.1039	0.2778	1.0263	0.4691	1.3621
10^{-8}	100	A	1.0617	2.917	1.2359	1.9953	1.1749
		AD	+0.26	+4.64	+0.92	+3.00	+0.70
		B	6.3879	4.8264	5.8621	3.5950	5.9854
		C	3.3728	0.1733	2.2816	0.4221	2.9219
10^{-10}	1	A	0.0337	0.0796		0.0590	
		AD	-14.73	-10.99	See	-12.29	See
		B	4.6590	3.1868	Case 1	3.2712	Case 3
		C	3.0079	0.2670		0.4598	

Table B-1 (continued)

P_{FA}	N	Coef.	0	1	2	3	4
			Distribution Case				
10^{-10}	10	A	0.2317	0.5164	0.2897	0.3724	0.2786
		AD	-6.35	-2.87	-5.38	-4.29	-5.55
		B	5.6485	3.4370	4.6576	3.3817	5.1942
		C	2.6065	0.2460	0.9326	0.4504	1.1766
10^{-10}	100	A	1.064	2.2803	1.1535	1.7418	1.1588
		AD	+0.27	+3.58	+0.62	+2.41	+0.64
		B	6.9134	3.4561	6.8119	3.6463	7.1206
		C	2.8617	0.2487	1.7367	0.4127	1.8606

APPENDIX C

Curve Fitting P_D vs SNR Curves

by E. N. Khoury

TSC Memorandum TSC-W21-36

TSC-W21-36/rad
March 6, 1979
A2095111

Curve Fitting P_d vs. SNR Curves

by E. N. Khoury

The probability of detection (P_d) for a given signal-to-noise ratio (S) is a complex function of the target mean cross section, target fluctuation statistics, number of pulses integrated and probability of false alarm (detection threshold setting). Computing this function requires evaluating some rather complicated integrals with limits which may extend to infinity. While this may be accomplished by using a Simpson's rule integrator or other such techniques in an iterative manner, these techniques are very general and provide accuracy at the expense of excessive use of computer time and core storage requirements. To reduce these core and time requirements on the program, an approximation method has been developed. This was done by computing a fit to the P_d vs. SNR case desired and using this fit in the Statistical Report Generator. The curve used to fit the data points is a three parameter low pass filter function of the form

$$\hat{P}_d(S) = 1 - \frac{1}{[k_1 * S^{k_2} + 1]^{k_3}} \qquad (1)$$

where $\hat{P}_d(S)$ is the estimate of P_d for any given value of SNR

and k_1, k_2, k_3 are the parameters which are varied to obtain the best fit to the desired model; i.e., Swerling Case I, M pulses integrated and $P_{fa} = 10^{-6}$.

If the cost function is selected as the squared-error criteria, the function to be minimized is

$$\begin{aligned} e^2(P_d) &= \sum_{i=1}^{N} \{P_{di}(S_i) - \hat{P}_{di}(S_i)\}^2 \\ &= \sum_{i=1}^{N} \{P_{di}(S_i) - [1 - k_1 S_i^{k_2} + 1)^{-k_3}]\}^2 \end{aligned} \qquad (2)$$

where

N	is the number of data points selected from the appropriate curve of P_d vs. SNR. Note that these points should cover the entire range of P_d values.
$P_{di}(S_i)$	is the true value of P_d for S_i selected from the appropriate curve.
$\hat{P}_{di}(S_i)$	is the estimate of $P_{di}(S)$.
k_1, k_2, k_3	are the model parameters.

The minimization is accomplished in a standard manner by taking the partial derivatives of the squared-error with respect to the three parameters and equating to zero. These partial derivatives are:

$$\frac{\partial[e^2(P_d)]}{\partial k_1} = -2\, k_3 \sum_{i=1}^{N} P_{di} S_i^{k_2} [k_1\, S_i^{k_2} + 1]^{-k_3-1}$$

$$+\ 2\, k_3 \sum_{i=1}^{N} S_i^{k_2}\, [k_1\, S_i^{k_2} + 1]^{-k_3-1}$$

$$-\ 2\, k_3 \sum_{i=1}^{N} S_i^{k_2}\, [k_1 S_i^{k_2} + 1]^{-2\, k_3-1} = 0 \qquad (3)$$

$$\frac{\partial[e^2(P_d)]}{\partial k_2} = -2\, k_1 k_3 \sum_{i=1}^{N} P_{di}\, \ln(S_i)\, S_i^{k_2}\, [k_1 S_i^{k_2} + 1]^{-k_3-1}$$

$$+\ 2\, k_1 k_3 \sum_{i=1}^{N} \ln\, (S_i)\, S_i^{k_2}\, [k_1 S_i^{k_2} + 1]^{-k_3-1}$$

$$-\ 2\, k_1 k_3 \sum_{i=1}^{N} \ln\, (S_i)\, S_i^{k_2}\, [k_1 S_i^{k_2} + 1]^{-2\, k_3-1} = 0 \qquad (4)$$

$$\frac{\partial[e^2(P_d)]}{\partial k_3} = -2 \sum_{i=1}^{N} (\ln[k_1 S_i^{k_2} + 1]) (k_1 S_i^{k_2} + 1)^{-k_3} P_{di}$$

$$+ \quad 2 \sum_{i=1}^{N} (\ln[k_1 S_i^{k_2} + 1]) (k_1 S_i^{k_2} + 1]^{-k_3}$$

$$- \quad 2 \sum_{i=1}^{N} (\ln[k_1 S_i^{k_2} + 1]) (k_1 S_i^{k_2} + 1)^{-2k_3} = 0 \qquad (5)$$

The required solution is then found by solving the Equations (3), (4) and (5) simultaneously for the three parameters k_1, k_2, and k_3.

More insight into how to iteratively determine a solution for equations (3), (4), and (5) may be obtained by rewriting these equations as follows:

$$\frac{\partial e^2(P_d)}{\partial k_i} = 2 \sum_{i=1}^{N} \{P_{di}(S_i) - \hat{P}_{di}(S_i)\} \frac{\partial P_{di}(S_i)}{\partial k_i}, \quad i = 1,2,3 \qquad (6)$$

where

$$\frac{\partial P_{di}}{\partial k_1} = -k_3 \, S_i^{k_2} \, [k_1 S_i^{k_2} + 1]^{-k_3-1} \qquad (7)$$

$$\frac{\partial P_{di}}{\partial k_2} = -k_1 k_3 \ln(S_i) \, S_i^{k_2} \, [k_1 S_i^{k_2} + 1]^{-k_3} \qquad (8)$$

$$\frac{\partial P_{di}}{\partial k_2} = -(\ln[k_1 \, S_i^{k_2} + 1]) \, (k_1 S_i^{k_2} + 1)^{-k_3-1} \qquad (9)$$

Finally, these equations may be solved iteratively be realizing that at steady state

$$\frac{\partial[e^2(P_d)]}{\partial k_i} = \frac{d\,k_i}{dt} \qquad (10)$$

One implementation of the solution to Equations (3), (4) and (5) using (6) through (10) is shown in Figure 1.

The parameters and fits obtained for five cases used in this study are presented in Tables 1 thru 5. Note that the column labeled "PDF" is the fit.

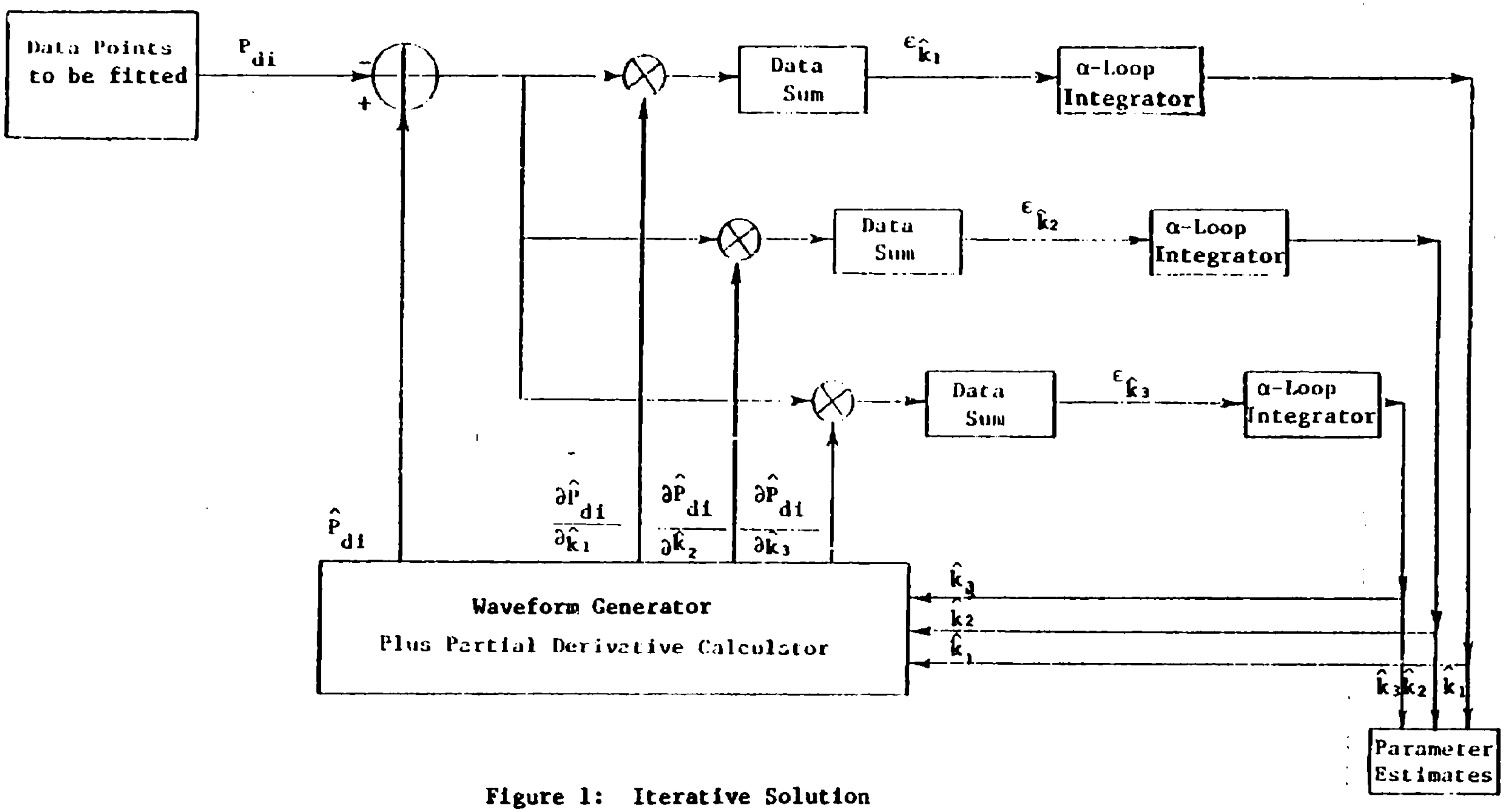

Figure 1: Iterative Solution to Equations (3), (4), and (5)

CASE NUMBER 1

Table 1

Parameter Fit for Swerling Case I, $P_{fa} = 10^{-6}$ and 1 Pulse Integrated

$$P_d = 1 - \frac{1}{\left(1 + k_1 S^{k_2}\right)^{k_3}}$$

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
1.0	0.1E-05	1

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
0.0118283	1.9788036	0.4340574

FIT

N	SNR	PD	PDF	ERROR
1	0.3820E 01	0.05695	0.06509	-0.00814
2	0.4950E 01	0.09815	0.10165	-0.00350
3	0.7744E 01	0.20609	0.20147	0.00462
4	0.1047E 02	0.30009	0.29460	0.00548
5	0.1379E 02	0.39197	0.38926	0.00271
6	0.1885E 02	0.49873	0.50054	-0.00181
7	0.2651E 02	0.60532	0.61007	-0.00474
8	0.3779E 02	0.70047	0.70494	-0.00447
9	0.6064E 02	0.79927	0.80022	-0.00095
10	0.1263E 03	0.89717	0.89272	0.00445
11	0.3265E 03	0.95870	0.95245	0.00625
12	0.1800E 04	0.99236	0.98903	0.00334

MEAN ERROR = 0.00027 RMS ERROR = 0.00459

CASE NUMBER 2

Table 2

Parameter Fit for Swerling Case I,

$P_{fa} = 10^{-6}$ and 4 Pulses Integrated

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
1.0	0.1E-05	4

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
0.0976332	2.0438943	0.4196889

FIT

N	SNR	PD	PDF	ERROR	[illegible]
1	0.1154E 01	0.04014	0.05030	-0.01016	-.52 dB
2	0.1719E 01	0.09999	0.10279	-0.00300	-.08
3	0.2600E 01	0.20247	0.19733	0.00514	+.08
4	0.3600E 01	0.30577	0.29987	0.00589	+.07
5	0.4661E 01	0.40911	0.40698	0.00214	+.02
6	0.6306E 01	0.49788	0.49979	-0.00191	-.02
7	0.8766E 01	0.60193	0.60692	-0.00499	-.07
8	0.1234E 02	0.69516	0.69999	-0.00483	-.08
9	0.2039E 02	0.80088	0.80185	-0.00098	-.03
10	0.4537E 02	0.90438	0.89951	0.00487	+.25
11	0.9718E 02	0.95402	0.94765	0.00637	+.66
12	0.5256E 03	0.99132	0.98769	0.00363	+1.77

MEAN ERROR = 0.00018 RMS ERROR = 0.00507

CASE NUMBER 3

Table 3

Parameter Fit for Swerling Case I.

$P_{fa} = 10^{-6}$ and 13 Pulses Integrated

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
1. 0.	0. 1E-05	13

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
0. 5393709	2. 1034484	0. 4062731

FIT

N	SNR	PD	PDF	ERROR
1	0. 5439E 00	0. 04538	0. 05505	-0. 00968
2	0. 7283E 00	0. 09040	0. 09438	-0. 00399
3	0. 1174E 01	0. 20985	0. 20415	0. 00569
4	0. 1535E 01	0. 29643	0. 29030	0. 00613
5	0. 2050E 01	0. 39697	0. 39450	0. 00247
6	0. 2751E 01	0. 49832	0. 50063	-0. 00231
7	0. 3787E 01	0. 60023	0. 60551	-0. 00529
8	0. 5283E 01	0. 69166	0. 69668	-0. 00502
9	0. 8519E 01	0 79650	0. 79753	-0. 00103
10	0. 1884E 02	0. 90057	0. 89553	0. 00504
11	0. 3975E 02	0. 95146	0. 94475	0. 00671
12	0. 2108E 03	0. 99064	0. 98672	0. 00393

MEAN ERROR = 0. 00022 RMS ERROR = 0. 00525

CASE NUMBER 4

Table 4

Parameter Fit for Swerling Case I.

$P_{fa} = 10^{-b}$ and 50 Pulses Integrated

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
1.0	0.1E-05	50

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
3.4545908	2.1581063	0.3971123

FIT

N	SNR	PD	PDF	ERROR
1	0.2503E 00	0.05209	0.06169	-0.00960
2	0.3396E 00	0.10032	0.10357	-0.00325
3	0.4811E 00	0.19749	0.19233	0.00515
4	0.6503E 00	0.29579	0.28952	0.00627
5	0.8520E 00	0.39514	0.39245	0.00269
6	0.1148E 01	0.49540	0.49752	-0.00211
7	0.1571E 01	0.59641	0.60165	-0.00525
8	0.2255E 01	0.69793	0.70290	-0.00497
9	0.3667E 01	0.79968	0.80067	-0.00099
10	0.7899E 01	0.90106	0.89615	0.00491
11	0.1647E 02	0.95117	0.94461	0.00656
12	0.8581E 02	0.99043	0.98654	0.00389

MEAN ERROR = 0.00027 RMS ERROR = 0.00512

CASE NUMBER 5

Table 5

Parameter Fit for Swerling Case I.

$P_{fa} = 10^{-6}$ and 100 Pulses Integrated

INPUT PARAMETERS

SWERLING CASE NO.	PFA	PULSES INTEGRATED
1.0	0.1E-05	100

PARAMETER ESTIMATES

PARAMETER 1	PARAMETER 2	PARAMETER 3
12.0662518	2.2491016	0.3787246

FIT

N	SNR	PD	PDF	ERROR
1	0.1443E 00	0.04214	0.05316	-0.01102
2	0.2069E 00	0.10471	0.10713	-0.00241
3	0.2996E 00	0.20587	0.19993	0.00593
4	0.4036E 00	0.30654	0.30039	0.00615
5	0.5337E 00	0.40691	0.40501	0.00190
6	0.7095E 00	0.50704	0.50996	-0.00292
7	0.9681E 00	0.60691	0.61244	-0.00552
8	0.1395E 01	0.70647	0.71100	-0.00453
9	0.2061E 01	0.79009	0.79102	-0.00093
10	0.4419E 01	0.89570	0.89030	0.00540
11	0.1090E 02	0.95628	0.94912	0.00717
12	0.5648E 02	0.99140	0.98747	0.00394

MEAN ERROR = 0.00026 RMS ERROR = 0.00547

APPENDIX D

Program PDFIT

```
C      PROGRAM PDFIT
C
C
$INSERT SYSCOM>KEYS.F
C
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
       INTEGER FILE6(3),CODE
       REAL CASES,PFAS,THS(3),SNRS(12),PDS(12),PDFS(12),
      1  ERRORS(12),ERRMS,SDERRS,SEC1,SEC2,SECS,SES
       INTEGER IAR(15)
C
       DIMENSION PD(12),ED(12),WK(3),ISTOP(12),ST(12),ERROR(12)
       DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
       DIMENSION IP(3),STP(3),NP(100)
C
       COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,KS
C
       DATA WK/0.5D0,0.5D0,0.5D0/
       DATA ST/0.001D0,0.001D0,0.001D0,0.001D0,0.001D0,0.001D0,
      10.001D0,0.001D0,0.001D0,0.001D0,0.001D0,0.001D0/
       DATA IPF/:100600/
C
C
C   PRINT INFORMATIVE MESSAGES.
C
       WRITE(1,3000)
       WRITE(1,3100)
C
C   ENTER INPUT DATA.
C
       WRITE(1,3200)
       READ(1,3300)FILE6
       WRITE(1,3400)
       READ(1,*)ICASE
       GO TO (1111,2222,3333), ICASE+1
1111   WRITE(1,3500)
       READ(1,*)PFAS
       GO TO 10
2222   WRITE(1,3600)
       READ(1,*)PFAS
       GO TO 10
3333   WRITE(1,3700)
       READ(1,*)PFAS
10     WRITE(1,3800)
       READ(1,*)NCASE
       WRITE(1,3900)
       READ(1,*)(NP(III),III=1,NCASE)
       WRITE(1,4000)
C
C   OPEN OUTPUT FILE.
C
       CALL SRCH$$(K$WRIT,FILE6,6,2,0,CODE)
       IF(CODE.NE.0) GO TO 9999
C
C   INTEGER/SINGULAR TO REAL/DOUBLE PRECISION CONVERSION.
C
       CASE=FLOAT(ICASE)
       PFA=DBLE(PFAS)
C
C   PROCESS SWERLING CASE NCASE TIMES.
C
       DO 999 ICNT=1,NCASE
C
C   PRINT CASE NUMBER BEING PROCSSED.
C
       WRITE(1,9000)ICNT
C
       NNP=NP(ICNT)
C
C   INITIALIZATION.
```

```
      C
            L=5
            M=0
            IP(1)=0
            KS=0
            IP(2)=0
            IP(3)=0
            ISS=0
            DEL(1)=0.0001D0
            DEL(2)=0.0001D0
            DEL(3)=0.0001D0
            STP(1)=0.000001D0
            STP(2)=0.000001D0
            STP(3)=0.000001D0
            ISUM=0
            DMIN=1.0D-2
            WEIGH=1.0D0
            SK=1.0D0
            DO 20 I=1,12
            ISTOP(I)=0
      20    CONTINUE
            SCASE=DBLE(FLOAT(NNP))
            IF(DABS(CASE-1.0D0).LE.0.1D0)SCASE=1.0D0
            IF(CASE-1.0D0.LE.-0.1D0) SCASE=1.1D9
      C
      C   COMPUTE ADAPTIVE INITIALIZATION FOR SWERLING CASE NUMBER 1.
      C
            RN=DBLE(FLOAT(NNP))
            IF(CASE.NE.1.0D0) GO TO 23
            TH(1)=0.11D0*RN**0.72D0
            TH(2)=2.12D0
            TH(3)=0.4D0
            GO TO 27
      C
      C   COMPUTE ADAPTIVE INITIALIZATION FOR SWERLING CASE NUMBER 2.
      C
      23    CONTINUE
            TH(1)=0.11D0*(DBLE(FLOAT(NNP)))**0.56D0
            DEV11=-DLOG10(PFA)-6.0D0
            SG=DSIGN(1.0D0,DEV11)
            D2P=SG*(0.01D0)**(1.0D0/RN)*(DABS(DEV11))**(1.0D0-1.0D0/RN)/2.0D0
            D3P=-0.8D0*D2P
            TH(2)=1.979D0+3.0D0*((RN-1.0D0)/RN)**6.0D0
            TH(2)=TH(2)+0.8D0*D2P
            TH(3)=0.434D0+1.56D0*((RN-1.0D0)/RN)**6.0D0
            IF(SG.LT.0.0D0) TH(3)=TH(3)+D3P
            IF((CASE.LE..1D0).AND.(NNP.LE.6))TH(2)=TH(2)+1.0D0
            IF((CASE.LE..1D0).AND.(NNP.LE.6))TH(3)=TH(3)+1.8D0
            IF((CASE.LE..1D0).AND.(NNP.LE.6))TH(1)=TH(1)/1.5D0
      27    PD(1)=0.05D0
            PD(2)=0.1D0
            PD(3)=0.2D0
            PD(4)=0.3D0
            PD(5)=0.4D0
            PD(6)=0.5D0
            PD(7)=0.6D0
            PD(8)=0.7D0
            PD(9)=0.8D0
            PD(10)=0.9D0
            PD(11)=0.95D0
            PD(12)=0.99D0
            SECS=0.
            SES=0.
      C
      C   COMPUTE SNR.
      C
            SNRMIN=-18.0D0
            SNRMAX=35.0D0
            DO 100 I=1,12
            T=(2.0D0*PD(I)-1.0D0)/1.11D0
            QD=1.231D0*T/DSQRT(1.0D0-T*T)
            Q=2.36D0*DSQRT(-DLOG10(PFA))-1.02D0
```

```
      XO=(G+GD)**2
      SN=XO/4.0D0/DBLE(FLOAT(NNP))*(1.0D0+DSQRT(1.0D0+16.0D0*
     1   DBLE(FLOAT(NNP))/XO))
      IF(CASE.EQ.0.0D0) SLF=1.0D0
      IF(CASE.EQ.0.0D0) GO TO 40
      F1=(1.0D0+GD/G)*(-DLOG(PD(I)))
      IF(CASE.EQ.1.0D0) GO TO 30
      SLF=1.0D0/(F1)**(1.0D0/DBLE(FLOAT(NNP)))
      GO TO 40
30    SLF=1.0D0/F1
40    SN=SN*SLF
      SN=10.0D0*DLOG10(SN)
      SNRMIN=SN-2.0D0
      SNRMAX=SN+2.0D0
      DO 80 J=1,10000
      SN=(SNRMIN+SNRMAX)/2.0D0
      CALL TIMDAT(IAR,15)
      SEC1=FLOAT(IAR(7))+FLOAT(IAR(8))/FLOAT(IAR(11))
      CALL DETECT (NNP,FFA,SN,SCASE,1.0D-4,PROB)
      CALL TIMDAT(IAR,15)
      SEC2=FLOAT(IAR(7))+FLOAT(IAR(8))/FLOAT(IAR(11))
      SECS=SECS+(SEC2-SEC1)
      D=PROB-PD(I)
      IF(D)50,60,70
50    IF(DABS(D).LE.DMIN) GO TO 90
      SNRMIN=SN
      GO TO 80
60    GO TO 90
70    IF(DABS(D).LE.DMIN) GO TO 90
      SNRMAX=SN
80    CONTINUE
90    SNR(I)=10.0D0**(0.1D0*SN)
      PD(I)=PROB
      SES=SES+SECS/FLOAT(J)
      SNRMAX=35.0D0
100   CONTINUE
      SES=SES/FLOAT(I)
      TH(1)=1.D0/TH(1)
      IF((CASE.GE..9D0).AND.(CASE.LE.1.1D0)) TH(1)=SNR(3)
      IF(CASE.LE..1D0) TH(1)=SNR(9)
C
C
C   COMPUTE INITIAL DECUPLING MATRIX.
      CALL VBAR
      CALL DCPLE
      DO 160 KS=1,1000
      M=M+1
      CALL VBAR
C
C   COMPUTE DECUPLING MATRIX.
C
      IF(M.GT.L) CALL DCPLE
C
C   FORM ERRORS.
C
      DO 110 I=1,12
      ED(I)=(PDF(I)-PD(I))*WEIGH
      IF(DABS(ED(I)).LE.ST(I))ISTOP(I)=1
      ISUM=ISUM+ISTOP(I)
110   CONTINUE
C
C   COMPUTE WEIGHTED ERRORS.
C
      DO 130 I=1,3
      DINC(I)=0.0D0
      DO 120 K=1,12
      DINC(I)=DINC(I)+W(I,K)*ED(K)
120   CONTINUE
C
C   COMPUTE UPDATED PARAMETER ESTIMATE.
C
      TH(I)=TH(I)-WK(I)*DINC(I)/SK
```

```
      SS=WK(I)*DINC(I)
      IF(DABS(SS).LE.STP(I))IP(I)=1
      ISS=ISS+IP(I)
C
C   CHECK IF ESTIMATE HAS CONVERGED.  IF YES, SET ISTOP(I)
C   EQUAL TO 1.
C
130   CONTINUE
C
C   HAVE ALL THREE ESTIMATES CONVERGED?
C
      IF(ISUM.EQ.12) GO TO 170
      IF(ISS.EQ.3) GO TO 170
      DO 140 I=1,12
      ISTOP(I)=0
140   CONTINUE
      DO 150 I=1,3
      IP(I)=0
150   CONTINUE
      ISS=0
      ISUM=0
160   CONTINUE
170   CONTINUE
      S2=0.0D0
      SS=0.0D0
      DO 180 I=1,12
      ERROR(I)=PD(I)-PDF(I)
      S2=S2+ERROR(I)**2
      SS=SS+ERROR(I)
180   CONTINUE
      ERRM=SS/12.0D0
      SDERR=DSQRT(S2/12.0D0-ERRM*ERRM)
      TH(1)=1.D0/TH(1)
C
C   DOUBLE PRECISION TO SINGLE PRECISION CONVERSION.
C
      CASES=SNGL(CASE)
      PFAS=SNGL(PFA)
      DO 190 I=1,3
      THS(I)=SNGL(TH(I))
190   CONTINUE
      DO 200 I=1,12
      SNRS(I)=SNGL(SNR(I))
      PDS(I)=SNGL(PD(I))
      PDFS(I)=SNGL(PDF(I))
      ERRORS(I)=SNGL(ERROR(I))
200   CONTINUE
      ERRMS=SNGL(ERRM)
      SDERRS=SNGL(SDERR)
C
C   WRITE TO OUTPUT FILE.
C
      WRITE(6,1400)ICNT
      WRITE(6,1420)
      WRITE(6,1440)
      WRITE(6,1460)
      WRITE(6,1500)
      WRITE(6,1600)
      WRITE(6,1700)CASES,PFAS,NNP
      WRITE(6,1800)
      WRITE(6,1900)
      WRITE(6,2000)
      WRITE(6,2100)
      WRITE(6,2200)THS(1),THS(2),THS(3)
      WRITE(6,2300)
      WRITE(6,2400)
      WRITE(6,2500)
      WRITE(6,2600)
      WRITE(6,2700)(I,SNRS(I),PDS(I),PDFS(I),ERRORS(I),I=1,12)
      WRITE(6,2800)ERRMS,SDERRS
C
C   GO TO TOP OF PAGE AND OBTAIN NEXT CASE.
```

```
C
      WRITE(6,2900)IPF
C
999   CONTINUE
C
C  REWIND  OUTPUT FILE.
C
      REWIND 6
C
C  CLOSE  OUTPUT FILE.
C
      CALL SRCH$$(K$CLOS,0,0,2,0,CODE)
      IF(CODE.NE.0) GO TO 9999
C
      GO TO 99999
C
C  FORMATS.
C
1400  FORMAT(//2X,'CASE NUMBER',I4)
1420  FORMAT(2X,'---- ------ ---')
1440  FORMAT(////36X,'INPUT PARAMETERS')
1460  FORMAT(36X,'----- ----------')
1500  FORMAT(//15X,'SWERLING CASE NO.',10X,'PFA',12X,
     1   'PULSES INTEGRATED')
1600  FORMAT(15X,'-------- ---- ---',10X,'---',12X,'------ ----------')
1700  FORMAT(//22X,F3.1,11X,E11.1,17X,I3)
1800  FORMAT(////34X,'PARAMETER ESTIMATES')
1900  FORMAT(34X,'--------- ---------')
2000  FORMAT(//18X,'PARAMETER 1',10X,'PARAMETER 2',10X,'PARAMETER 3')
2100  FORMAT(18X,'--------- -',10X,'--------- -',10X,'--------- -')
2200  FORMAT(//14X,F14.7,7X,F14.7,7X,F14.7)
2300  FORMAT(////42X,'FIT')
2400  FORMAT(42X,'---')
2500  FORMAT(//18X,'N',10X,'SNR',10X,'PD',10X,'PDF',10X,'ERROR')
2600  FORMAT(18X,'-',10X,'---',10X,'--',10X '---',10X,'-----')
2700  FORMAT(17X,I2,6X,E11.4,3X,F8.5,4X,F8.5,6X,F8.5)
2800  FORMAT(///19X'MEAN ERROR = ',F7.5,12X,'RMS ERROR = ',F7.5)
2900  FORMAT(A2)
3000  FORMAT(///7X,'PROGRAM PDFIT COMPUTES AN EMPIRICAL FIT TO THE'/,
     1   2X,' EXACT PD VS. SNR CURVE DESIRED. THE EXACT CURVE IS'/,
     1   2X,' SPECIFIED BY SWERLING CASE NUMBER, PROBABLITY OF'/,
     1   2X,' FALSE ALARM, AND NUMBER OF PULSES NON-COHERENTLY'/,
     1   2X,' INTEGRATED. THIS PROGRAM WILL RETURN THE THREE'/,
     1   2X,' PARAMETERS WHICH GIVE THE BEST FIT IN THE MMSE'/,
     1   2X,' SENSE (CF. REF. TSC-W21-40)'/,
     1   6X,' PDFIT WAS DEVELOPED BY E. N. KHOURY, AND R. F.'/,
     1   2X,' PUSATERI - JUNE 79.'///)
3100  FORMAT(3X,'INPUT DATA CONSISTS OF THE FOLLOWING:'//,
     1   3X,'     A) CUTPUT FILE NAME'/,
     1   3X,'     B) SWERLING CASE NUMBER'/,
     1   3X,'     C) PROBAALITY OF FALSE ALARM'/,
     1   3X,'     D) NUMBER OF CASES TO BE RUN'/,
     1   3X,'     E) NUMBER OF PULSES INTEGRATED FOR EACH CASE')
3200  FORMAT(//3X,'ENTEP OUTPUT FILE NAME - 6')
3300  FORMAT(3A2)
3400  FORMAT(//3X,'ENTER SWERLING CASE NUMBER - 0 1 2')
3500  FORMAT(//3X,'ENTER PROBABLITY OF FALSE ALARM'/,
     1 '      RANGE: 1.E-3 THROUGH 1.E-15')
3600  FORMAT(//3X,'ENTER PROBABLITY OF FALSE ALARM - PFA'/,
     1 '      RANGE: 1.E-2 THROUGH 1.E-38')
3700  FORMAT(//3X,'ENTER PROBABLITY OF FALSE ALARM - PFA'/,
     1 '      RANGE: 1.E-3 THROUGH 1.E-8')
3800  FORMAT(//3X,'ENTER NUMBER OF CASES TO BE RUN'/,
     1 '      RANGE: 1 THROUGH 100')
3900  FORMAT(//3X,'ENTER NUMBER OF INTEGRATED PULSES FOR EACH CASE'/,
     1 '      RANGE: 1 THROUGH 100')
4000  FORMAT(//)
9000  FORMAT('PROCESSING CASE NUMBER: ',I4)
C
9901  FORMAT(/2X,D25.8,5X,D25.8,5X,D25.8)
9902  FORMAT(/2X,I2,5X,D25.8)
9903  FORMAT(/2X,I3,5X,I2,5X,D25.8)
```

```
C
9999  CALL ERRPR$(K$NRTN,CODE,0,0,'PDFIT',5)
C
C
99999 CALL EXIT
      END
```

```
      SUBROUTINE DCPLE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION PS(12),TS(12),A(3,3),B(3,3),DE(3,12)
      DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
C
      COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,II
C
C
      WEIGH=1.D0
      DO 4 I=1,3
      DO 2 J=1,3
      A(I,J)=0.
2     CONTINUE
4     CONTINUE
      DO 10 I=1,12
      PS(I)=PDF(I)
10    CONTINUE
      DO 30 I=1,3
      TS(I)=TH(I)
      TH(I)=TH(I)+DEL(I)
      CALL VBAR
      DO 20 J=1,12
      DE(I,J)=(PDF(J)-PS(J))/DEL(I)/WEIGH
20    CONTINUE
      TH(I)=TS(I)
30    CONTINUE
      DO 60 MK=1,3
      DO 50 I=1,3
      DO 40 J=1,12
      A(MK,I)=DE(MK,J)*DE(I,J)+A(MK,I)
      W(I,J)=0.
40    CONTINUE
50    CONTINUE
60    CONTINUE
      CALL MXINVR(A,B,IBAD)
      IF(IBAD.EQ.1) WRITE(1,1000)
      DO 90 MK=1,3
      DO 80 J=1,12
      DO 70 I=1,3
      W(MK,J)=W(MK,J)+B(MK,I)*DE(I,J)
70    CONTINUE
80    CONTINUE
90    CONTINUE
      DO 100 I=1,12
      PDF(I)=PS(I)
100   CONTINUE
      GO TO 108
105   CONTINUE
      DO 107 I=1,3
      DO 106 J=1,12
106   W(I,J)=DE(I,J)
107   CONTINUE
108   CONTINUE
      M=0
C
1000  FORMAT('DCPLE: ** ERROR ** IBAD = 1')
C
      RETURN
      END
```

```
      SUBROUTINE DETECT(N,PFA,SN,SWERL,TOL,PROB)
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C
      T=THRESH(PFA,N,32)
      PROB=PROBDE(T,SN,SWERL,32,N,TOL)
C
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION GFUNCT(T,N,M)
C
C
C  THIS FUNCTION COMPUTES THE INCOMPLETE
C  GAMMA FUNCTION.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/DETPD/ TO,TT,G,C,R,D
C
C
      TO=T
      IF(M.GT.20) GO TO 30
      R=M*N
      TT=T/DBLE(FLOAT(M))
      C=1./(TT+1.)**R
      D=TT/(1.+TT)
      G=0.D0
      DO 20 K=1,N
      G=G+C
      C=C*(R+DBLE(FLOAT(K))-1.)*D/DBLE(FLOAT(K))
20    CONTINUE
      GO TO 50
30    TT=T*DBLE(FLOAT(N))
      C=DEXP(-TT)
      G=0.D0
      DO 40 K=1,N
      G=G+C
      C=C*TT/DBLE(FLOAT(K))
40    CONTINUE
50    GFUNCT=G
C
      RETURN
      END
```

```
      1   SEQ NEST       SUBROUTINE MXINVR(A,B,IBAD)

           1             SUBROUTINE MXINVR(A,B,IBAD)

           2         C  SUBROUTINE MXINVR FACTORS MATRIX A INTO LU CROUT DECOMPOSITITON
           3         C  AND COMPUTES ITS INVERSE, MATRIX B.

           4             IMPLICIT DOUBLE PRECISION (A-H,O-Z)

           5         C

           6             DIMENSION A(3,3),B(3,3)

           7         C

           8             FOR(I=1 TO 3)
           9    1        .    FOR(J=1 TO 3)
          10    2        .    .    B(I,J)=0.
          11    2        .    .    IF(I.EQ.J) B(I,J)=1.
          12    1        .    END FOR
          13             END FOR

          14         C

          15             IBAD=0
          16             FOR(K=1 TO 3)
          17    1        .    FOR(I=K TO 3)
          18    2        .    .    DINN1=0.D0
          19    2        .    .    FOR(L=1 TO (K-1))
          20    3        .    .    .    DINN1=DINN1+A(I,L)*A(L,K)
          21    2        .    .    END FOR
          22    2        .    .    A(I,K)=A(I,K)-DINN1
          23    1        .    END FOR

          24    1    C      *    TEST FOR SINGULARITY.

          25    1        .    IF(A(K,K).NE.0)
          26    2        .    .    FOR(J=(K+1) TO 3)
          27    3        .    .    .    DINN2=0.D0
          28    3        .    .    .    FOR(L=1 TO (K-1))
          29    4        .    .    .    .    DINN2=DINN2+A(K,L)*A(L,J)
          30    3        .    .    .    END FOR
          31    3        .    .    .    A(K,J)=(A(K,J)-DINN2)/A(K,K)
          32    2        .    .    END FOR
          33    1        .    ELSE
          34    2        .    .    IBAD=1
          35    2        .    .    K=4
          36    1        .    END IF
          37             END FOR

          38         C

          39             IF(IBAD.NE.1)
          40    1        .    FOR(M=1 TO 3)
          41    2        .    .    FOR(I=1 TO 3)
          42    3        .    .    .    DINN1=0.D0
          43    3        .    .    .    FOR(K=1 TO (I-1))
          44    4        .    .    .    .    DINN1=DINN1+A(I,K)*B(K,M)
      1   SEQ NEST       SUBROUTINE MXINVR(A,B,IBAD)

          45    3        .    .    .    END FOR
          46    3        .    .    .    B(I,M)=(B(I,M)-DINN1)/A(I,I)
          47    2        .    .    END FOR
          48    2        .    .    FOR(I=3 TO 1 STEP -1)
          49    3        .    .    .    DINN1=0.D0
          50    3        .    .    .    FOR(J=(I+1) TO 3)
          51    4        .    .    .    .    DINN1=DINN1+A(I,J)*B(J,M)
          52    3        .    .    .    END FOR
          53    3        .    .    .    B(I,M)=B(I,M)-DINN1
          54    2        .    .    END FOR
          55    1        .    END FOR
          56             ELSE
```

```
      57   1              RETURN
      58              END IF

      59       C

      60              RETURN
      61              END

IFTRAN STATISTICS
      61 CARDS READ
       0 ERROR(S) FOUND
```

```
      DOUBLE PRECISION FUNCTION PROBDE(T,XDB,AK,M,N,ERR)
C
C
C  FUNCTION CALCULATES PROBABILITY OF DETECTION FOR
C  GIVEN THRESHOLD T, CASE AK, NUMBER OF PULSESN,
C  NUMBER OF CFAR CELLS M, AND SIGNAL-TO-NOISE XDB.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      INTEGER*4 J
C
      COMMON /DETPD/ TO,TT,GO,CO,R,D
C
      DATA TO/-1.0D0/
C
C
      IF(DABS(AK-DBLE(FLOAT(N))).GT.0.1D0) GO TO 40
      PROBDE=GFUNCT(T/(1.D0+10.D0**(.1D0*XDB)),N,M)
C
      RETURN
C
40    IF(T.NE.TO) G=GFUNCT(T,N,M)
      G=GO
      C=CO
      XN=DBLE(FLOAT(N))*10.D0**(.1D0*XDB)
      IF(AK.LE.1000.D0) GO TO 41
C
C  CHECK FOR POSSIBLE UNDERFLOW.
C
      IF(XN.GT.700.D0) GO TO 60
      A=DEXP(-XN)
      GO TO 42
41    A=(1.D0+XN/AK)**(-AK)
42    E=1.D0-A
      J=0
      PD=A*G
50    A=A*XN/(1.D0+DBLE(FLOAT(J)))
      IF(AK.LE.1000.D0) A=A*(1.D0+(DBLE(FLOAT(J)))/AK)/(1.D0+XN/AK)
      J=J+1
      IF(FLOAT(J).GT.1.E6) GO TO 53
      G=G+C
C
C  IF M.GT.20, TREAT AS IDEAL CFAR.
C
      IF(M.GT.20) GO TO 51
      C=C*(R+DBLE(FLOAT(N+(FLOAT(J))-1)))*D/DBLE(FLOAT(N+(FLOAT(J))))
      GO TO 52
51    C=C*TT/DBLE(FLOAT(N+(FLOAT(J))))
52    PD=PD+A*G
      E=E-A
      IF(E.GT.ERR) GO TO 50
      PROBDE=PD
C
      RETURN
C
53    CONTINUE
      PROBDE=PD
C
      RETURN
C
C  PRINT ERROR MESSAGE - UNDERFLOW.
C
60    WRITE(1,1000)
C
      STOP
C
C
1000  FORMAT(/2X,33HUNDERFLOW FOR NONFLUCTUATING CASE)
C
      END
```

```
      DOUBLE PRECISION FUNCTION THRESH(PFA,N,M)
C
C
C     THIS FUNCTION COMPUTES THE DETECTION THRESHOLD
C     FOR A GIVEN PROBABILITY OF FALSE ALARM PFA,
C     NUMBER OF PULSES N, AND NUMBER OF CFAR CELLS M.
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C
      ERR=PFA/100.DO
      TO=1.
      T1=1.1
      FO=GFUNCT(TO,N,M)-PFA
      DO 20 I=1,1000
      F1=GFUNCT(T1,N,M)-PFA
      IF(DABS(F1).LT.ERR) GO TO 30
      T2=T1-F1*(T1-TO)/(F1-FO)
      TO=T1
      T1=T2
      FO=F1
20    CONTINUE
30    THRESH=T1
C
      RETURN
      END
```

```
      SUBROUTINE VBAR
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
C
      COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,II
C
C
      DO 10 I=1,12
      PDF(I)=1.0+((1.D0/TH(1))*SNR(I))**TH(2)
      IF(PDF(I).LT.1.D0)PDF(I)=1.D0
      PDF(I)=1.0/PDF(I)**TH(3)
      PDF(I)=1.-PDF(I)
   10 CONTINUE
C
      RETURN
      END
```

APPENDIX E

Program QFIT

```
C      PROGRAM QFIT
C
       DIMENSION S(50),SDB(50),P(50),PFIT(50),ERR(50),RELERR(50)
       DIMENSION SDBW(15),W(50),AK(3),PDW(50),TH(3)
C
       PF(A1,B1,C1,X)=1.-(1.+(A1*X)**B1)**(-C1)
C
C      GET INPUT DATA.
C
       CALL QFIN(N,SDB,P)
C
C      CONVERT FROM DB TO POWER RATIO.
C
       DO 50 I=1,N
           S(I)=10.**(SDB(I)*.1)
50     CONTINUE
C
C      ESTIMATE COEFFICIENTS.
C
       WRITE(1,1002)
       READ(1,*)IUSE
       IF(IUSE .NE. 1)GO TO 55
       CALL ESTCOF(N,P,SDB,AK)
       GO TO 65
55     DO 60 I=1,N
           PDW(I)=P(I)*100.
           SDBW(I)=SDB(I)
60     CONTINUE
       CALL QPDFIT(PDW,SDBW,N,AK)
65     CONTINUE
C
C      COMPUTE FITED VALUES OF PD.
C
       DO 100 I=1,N
           PFIT(I)=PF(AK(1),AK(2),AK(3),S(I))
100    CONTINUE
C
C      COMPUTE ERRORS.
C
       CALL ERRORS(P,PFIT,N,ERR,RELERR,AVERR,AVREL)
C
C      OUTPUT PRINTED RESULTS.
C
       CALL QFOUT(SDB,S,P,PFIT,ERR,N,AK)
C
C      PLOT RESULTS IF DESIRED.
C
       WRITE(1,1001)
       READ(1,*)IPLOT
       IF(IPLOT .EQ. 1)CALL FTPLOT(N,S,P,AK)
C
C      DO COMPLETE PDFIT IF DESIRED.
C
       WRITE(1,1003)
       READ(1,*)IPDF
       IF(IPDF .NE. 1)GO TO 300
           TH(1)=AK(1)
           TH(2)=AK(2)
           TH(3)=AK(3)
           CALL PDFIT(N,SDB,P,TH)
C
C          PLOT RESULTS IF DESIRED.
C
           WRITE(1,1001)
           READ(1,*)IPLOT
           IF(IPLOT .EQ. 1)CALL FTPLOT(N,S,P,TH)
300    CONTINUE
C
       CALL EXIT
1001   FORMAT('IF A PLOT OF RESULTS IS DESIRED ENTER 1, ELSE 0')
1002   FORMAT('TO USE BOBS VERSION ENTER 1,ELSE 0')
1003   FORMAT('TO DO COMPLETE PDFIT ENTER 1, ELSE 0')
       END
```

```
A1       R           000000  0006
AK       R           000002  0004S  0023A  0029A  0035A  0044A  0050A  0057
                             0058   0059
AVERR    R           002155  0040A
AVREL    R           002157  0040A
B1       R           000000  0006
C1       R           000000  0006
ERR      R           000010  0003S  0040A  0044A
ERRORS   R EXTERNAL  000000  0040
ESTCOF   R EXTERNAL  000000  0023
EXIT     R EXTERNAL  000000  0069
FTPLOT   R EXTERNAL  000000  0050   0066
I        I           002161  0014M  0015   0025M  0026   0027   0034M  0035
IPDF     I           002163  0055M  0056
IPLOT    I           002164  0049M  0050   0065M  0066
IUSE     I           002165  0021M  0022
N        I           002166  0010A  0014   0023A  0025   0029A  0034   0040A
                             0044A  0050A  0060A  0066A
P        R           000154  0003S  0010A  0023A  0026   0040A  0044A  0050A
                             0060A  0066A
PDFIT    R EXTERNAL  000000  0060
PDW      R           000320  0004S  0026M  0029A
PF       R           001514  0006S  0035
PFIT     R           000464  0003S  0035M  0040A  0044A
QFIN     R EXTERNAL  000000  0010
QFOUT    R EXTERNAL  000000  0044
QPDFIT   R EXTERNAL  000000  0029
RELERR   R           000630  0003S  0040A
S        R           000774  0003S  0015M  0035A  0044A  0050A  0066A
SDB      R           001140  0003S  0010A  0015   0023A  0027   0044A  0060A
SDBW     R           001304  0004S  0027M  0029A
TH       R           001342  0004S  0057M  0058M  0059M  0060A  0066A
W        R           001350  0004S
X        R           000000  0006

_100                 001704  0034   0036D
_1001                002050  0048   0064   0070D
_1002                002103  0020   0071D
_1003                002127  0054   0072D
_300                 002047  0056   0067D
_50                  001577  0014   0016D
_55                  001633  0022   0025D
_60                  001651  0025   0028D
_65                  001665  0024   0030D

0000 ERRORS [<.MAIN.>FTN-REV15.3]
```

```
      SUBROUTINE DCPLE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION PS(12),TS(12),A(3,3),B(3,3),DE(3,12)
      DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
C
      COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,II,JV
C
C
      WEIGH=1.D0
      DO 4 I=1,3
      DO 2 J=1,3
      A(I,J)=0.
2     CONTINUE
4     CONTINUE
      DO 10 I=1,JV
      PS(I)=PDF(I)
10    CONTINUE
      DO 30 I=1,3
      TS(I)=TH(I)
      TH(I)=TH(I)+DEL(I)
      CALL VBAR
      DO 20 J=1,JV
      DE(I,J)=(PDF(J)-PS(J))/DEL(I)/WEIGH
20    CONTINUE
      TH(I)=TS(I)
30    CONTINUE
      DO 60 MK=1,3
      DO 50 I=1,3
      DO 40 J=1,JV
      A(MK,I)=DE(MK,J)*DE(I,J)+A(MK,I)
      W(I,J)=0.
40    CONTINUE
50    CONTINUE
60    CONTINUE
      CALL MXINVR(A,B,IBAD)
      IF(IBAD.EQ.1) WRITE(1,1000)
      DO 90 MK=1,3
      DO 80 J=1,JV
      DO 70 I=1,3
      W(MK,J)=W(MK,J)+B(MK,I)*DE(I,J)
70    CONTINUE
80    CONTINUE
90    CONTINUE
      DO 100 I=1,JV
      PDF(I)=PS(I)
100   CONTINUE
      GO TO 108
105   CONTINUE
      DO 107 I=1,3
      DO 106 J=1,JV
106   W(I,J)=DE(I,J)
107   CONTINUE
108   CONTINUE
      M=0
C
1000  FORMAT('DCPLE: ** ERROR ** IBAD = 1')
C
      RETURN
      END
```

```
A        D              000002   0006S   0015M   0033M   0038A
B        D              000046   0006S   0038A   0043
DCPLE    D              000000   0001S
DE       D              000112   0006S   0026M   0033    0043    0054
DEL      D  /COM/       000154   0007S   0009S   0023    0026
DINC     D  /COM/       000170   0007S   0009S
I        I              001125   0013M   0015    0018M   0019    0021M   0022    0023
                                 0026    0028    0031M   0033    0034    0042M   0043
                                 0047M   0048    0052M   0054
IBAD     I              001131   0038A   0039
II       I  /COM/       000425   0009S
J        I              001132   0014M   0015    0025M   0026    0032M   0033    0034
                                 0041M   0043    0053M   0054
JV       I  /COM/       000426   0009S   0018    0025    0032    0041    0047    0053
M        I  /COM/       000424   0009S   0057M
MK       I              001133   0030M   0033    0040M   0043
MXINVR   I  EXTERNAL    000000   0038
PDF      D  /COM/       000060   0007S   0009S   0019    0026    0048M
PS       D              000332   0006S   0019M   0026    0048
SNR      D  /COM/       000000   0007S   0009S
TH       D  /COM/       000140   0007S   0009S   0022    0023M   0028M
TS       D              000412   0006S   0022M   0028
VBAR     D  EXTERNAL    000000   0024
W        D  /COM/       000204   0007S   0009S   0034M   0043M   0054M
WEIGH    D              001134   0012M   0026

_10                     000537   0018    0020D
_100                    001040   0047    0049D
_1000                   001102   0039    0059D
_105                    001046   0051D
_106                    001052   0053    0054D
_107                    001072   0052    0055D
_109                    001100   0050    0056D
_2                      000513   0014    0016D
_20                     000614   0025    0027D
_30                     000631   0021    0029D
_4                      000521   0013    0017D
_40                     000706   0032    0035D
_50                     000714   0031    0036D
_60                     000722   0030    0037D
_70                     001006   0042    0044D
_80                     001014   0041    0045D
_90                     001022   0040    0046D

0000 ERRORS [<DCPLE >FTN-REV15.3]
```

```
      SUBROUTINE ERRORS(XACT, XMES, N, ERR, RELERR, AVERR, AVREL, SDVERR)
C
C     DESCRIPTION
C     -----------
C     COMPUTES THE ABSOLUTE AND RELATIVE ERRORS FOR EACH DATA POINT.
C     ALSO COMPUTES THE AVERAGE ABSOLUTE ERROR, AVERAGE RELATIVE ERROR,
C     & THE STANDARD DEVIATION OF THE DATA.
C
C     METHOD
C     ------
C     USES THE STANDARD DEFINITIONS FOR ABSOLUTE, RELATIVE ERRORS,
C     & STANDARD DEVIATION.
C
C     VARIABLE DESCRIPTIONS
C     -------- ------------
C
C     NAME  TYPE  DIM     DESCRIPTION                         UNITS  VIA
C
C     INPUTS:
C
C     N       I     1     NUMBER OF DATA POINTS                 -    CALL SEQ
C     XACT    R     N     ACTUAL VALUES OF DATA                 -    CALL SEQ
C     XMES    R     N     MEASURED VALUES OF DATA               -    CALL SEQ
C
C     OUTPUTS:
C
C     AVERR   R     1     AVERAGE ABSOLUTE ERROR                -    CALL SEQ
C     AVREL   R     1     AVERAGE RELATIVE ERROR                %    CALL SEQ
C     ERR     R     N     ABSOLUTE ERROR OF EACH POINT          -    CALL SEQ
C     RELERR  R     N     RELATIVE ERROR OF EACH POINT          %    CALL SEQ
C     SDVERR  R     1     STANDARD DEVIATION OF DATA            -    CALL SEQ
C
C     IMPORTANT LOCAL VARIABLES: NONE
C
C     EXTERNAL REFERENCES
C     -------------------
C     NONE
C
C     REFERENCE BOOKS, REPORTS/MEMOS
C     ------------------------------
C
C
C     AUTHOR
C     ------
C     R. E. BLASE
C
      DIMENSION XACT(1), XMES(1), ERR(1), RELERR(1)
C
C     COMPUTE ERRORS FOR EACH COMPONENT.
C
      S1=0.
      S2=0.
      SSQ=0.
      IBAD=0
      DO 50 I=1, N
          ERR(I)=XACT(I)-XMES(I)
          IF(XACT(I) .EQ. 0.)IBAD=IBAD+1
          IF(XACT(I).NE.0.)RELERR(I)=(ERR(I)/XACT(I))*100.
          S1=S1+ERR(I)
          S2=S2+RELERR(I)
          SSQ=SSQ+ERR(I)*ERR(I)
50    CONTINUE
C
C     COMPUTE AVERAGE ERRORS.
C
      AVERR=S1/N
      NUSE=N-IBAD
      AVREL=S2/NUSE
C
C     COMPUTE STANDARD DEVIATION.
C
      SDVERR=SQRT(SSQ/N-AVERR*AVERR)
C
      RETURN
      END
```

```
AVERR   R  ARGUMENT 000010   0001S   0066M   0072
AVREL   R  ARGUMENT 000011   0001S   0068M
ERR     R  ARGUMENT 000006   0001S   0047S   0056M   0058    0059    0061
ERRORS  R           000000   0001S
I       I           000205   0055M   0056    0057    0058    0059    0060    0061
IBAD    I           000212   0054M   0057M   0067
N       I  ARGUMENT 000005   0001S   0055    0066    0067    0072
NUSE    I           000213   0067M   0068
RELERR  R  ARGUMENT 000007   0001S   0047S   0058M   0060
S1      R           000216   0051M   0059M   0066
S2      R           000220   0052M   0060M   0068
SDVERR  R  ARGUMENT 000012   0001S   0072M
SQRT    R  EXTERNAL 000000   0072
SSQ     R           000222   0053M   0061M   0072
XACT    R  ARGUMENT 000003   0001S   0047S   0056    0057    0058
XMES    R  ARGUMENT 000004   0001S   0047S   0056

_50                 000132   0055    0062D

0000 ERRORS [<ERRORS>FTN-REV15.3]
```

```
      SUBROUTINE ESTCOF(N,P,SDB,AK)
C
C     DESCRIPTION
C     -----------
C     COMPUTES THE 1ST ESTIMATE OF THE PARAMETERS FOR THE
C     PDFIT FUNCTION.
C
C     METHOD
C     ------
C     USES THE ALOGRITHM DEVELOPED BY J. BUCKNAM DESCRIBED IN
C      A MEMO ON 'QUICK PDFIT'.
C
C     VARIABLE DESCRIPTIONS
C     -------- ------------
C
C     NAME TYPE DIM   DESCRIPTION                      UNITS  VIA
C
C     INPUTS:
C     N      I    1   NUMBER OF (S,P) DATA PAIRS        -    CALL SEQ
C     P      R    N   ACTUAL PD VALUES                  -    CALL SEQ
C     SDB    R    N   ACTUAL SIGNAL-TO-NOISE RATIOS     DB   CALL SEQ
C
C     OUTPUTS:
C     AK     R    3   INITIAL ESTIMATES FOR K1,K2,K3 -       CALL SEQ
C
C     LOCAL VARIABLES:
C     A      R    1   SLOPE OF W VERS SDB FIT
C     B      R    1   Y INTERCEPT OF W VESR SDB FIT
C     AK3UN  R    1   UNCORRECTED ESTIMATE FOR K3
C     SRL    R    1   RATIO OF SLOPE OF TRUE FUNCTION TO SLOPE
C                     OF ASYMPTOTIC FUNCTION AT LARGE PD
C     SRS    R    1   RATIO OF SLOPE OF TRUE FUNCTION TO SLOPE
C                     OF ASYMPTOTIC FUNCTION AT SMALL PD
C     Y1     R    1   1ST VALUE USED FOR SMALL PD
C     Y2     R    1   2ND VALUE USED FOR SMALL PD
C     YM     R    1   SLOPE OF Y VERS. SDB CURVE
C     WI     R    1   VALUES OF BUCKNAM FUNCTION        -
C     Z3     R    1   1ST VALUE USED FOR LARGE PD
C     Z4     R    1   2ND VALUE USED FOR LARGE PD
C     ZM     R    1   SLOPE OF Z VERS. SDB CURVE
C
C     EXTERNAL REFS:
C     -------------
C     INSORT -INSERTION SORTING ROUTINE
C
C     REFERENCE BOOKS, REPORTS/MEMOS
C     ------------------------------
C     TSC-W7-73/RAD, J. BUCKNAM, 1979
C
C     AUTHOR
C     ------
C     R. E. BLASE
C
      DIMENSION P(1),SDB(1),AK(3)
C
C     SORT P & S ARRAYS ACCORDING TO INCREASING PD.
C
      CALL INSORT(SDB,P,1,N)
C
C     COMPUTE SLOPE FOR SMALL PD.
C
      Y1=ALOG10(-ALOG10(1.-P(1)))
      Y2=ALOG10(-ALOG10(1.-P(2)))
      YM=(Y2-Y1)/(SDB(2)-SDB(1))
C
C     COMPUTE SLOPE FOR LARGE PD.
C
      NM1=N-1
      Z3=-ALOG10(1.-P(NM1))
      Z4=-ALOG10(1.-P(N))
      ZM=(Z4-Z3)/(SDB(N)-SDB(NM1))
C
```

```
C     COMPUTE UNCORRECTED VALUE OF K3.
C
      AK3UN=ZM/YM
      RAK3UN=1./AK3UN
C
C     COMPUTE CORRECTION FACTOR.
C
      PTIL1=1.-(1.-P(1))**(RAK3UN)
      PTIL2=1.-(1.-P(2))**(RAK3UN)
      PTIL3=1.-(1.-P(NM1))**(RAK3UN)
      PTIL4=1.-(1.-P(N))**(RAK3UN)
      SRLT2=ALOG10(((1.-PTIL4)**(-RAK3UN))-1.)
      SRLT1=ALOG10(((1.-PTIL3)**(-RAK3UN))-1.)
      SRLB2=(-RAK3UN)*ALOG10(1.-PTIL4)
      SRLB1=(-RAK3UN)*ALOG10(1.-PTIL3)
      SRL=(SRLT2-SRLT1)/(SRLB2-SRLB1)
      SRST2=ALOG10(((1.-PTIL2)**(-RAK3UN))-1.)
      SRST1=ALOG10(((1.-PTIL1)**(-RAK3UN))-1.)
      CONST=ALOG(10.0)
      SRSB2=ALOG10((-CONST*RAK3UN)*(ALOG10(1.-PTIL2)))
      SRSB1=ALOG10((-CONST*RAK3UN)*(ALOG10(1.-PTIL1)))
      SRS=(SRST2-SRST1)/(SRSB2-SRSB1)
C
C     CORRECT K3.
C
      AK3=AK3UN*SRL/SRS
      RAK3=1./AK3
C
C     COMPUTE VALUES OF BUCKNAM FUNCTION FOR GIVEN PD'S.
C     FIT W TO W=A*S +B
C
      SUM=FLOAT(N)
      SUMX=0.
      SUMY=0.
      SUMX2=0.
      SUMXY=0.
      DO 200 I=1,N
          SDBI=SDB(I)
          WI=ALOG10(((1.-P(I))**(-RAK3))-1.)
          SUMX=SUMX+SDBI
          SUMY=SUMY+WI
          SUMX2=SUMX2+SDBI*SDBI
          SUMXY=SUMXY+SDBI*WI
200   CONTINUE
      D=SUM*SUMX2-SUMX*SUMX
      B=(SUMX2*SUMY-SUMX*SUMXY)/D
      A=(SUMXY*SUM-SUMX*SUMY)/D
C
C     COMPUTE K1,K2
C
      AK10=10.*A
      AK(1)=10.**(B/AK10)
      AK(2)=AK10
      AK(3)=AK3
C
      RETURN
      END
```

```
A      R          000747  0119M  0123
AK     R ARGUMENT 000006  0001S  0054S  0124M  0125M  0126M
AK10   R          000751  0123M  0124   0125
AK3    R          000753  0098M  0099   0126
AK3UN  R          000755  0075M  0076   0098
ALOG   R EXTERNAL 000000  0091
ALOG10 R EXTERNAL 000000  0062   0063   0069   0070   0084   0085   0086
                          0087   0089   0090   0092   0093   0111
B      R          000757  0118M  0124
CONST  R          000761  0091M  0092   0093
D      R          000763  0117M  0118   0119
ESTCOF R          000000  0001S
FLOAT  R EXTERNAL 000000  0104
I      I          000765  0109M  0110   0111
INSORT I EXTERNAL 000000  0053
N      I ARGUMENT 000003  0001S  0058A  0068   0070   0071   0083   0104
                          0109
NM1    I          000770  0068M  0069   0071   0082
P      R ARGUMENT 000004  0001S  0054S  0058A  0062   0063   0069   0070
                          0080   0081   0082   0083   0111
PTIL1  R          000771  0080M  0090   0093
PTIL2  R          000773  0081M  0089   0092
PTIL3  R          000775  0082M  0085   0087
PTIL4  R          000777  0083M  0084   0086
RAK3   R          001011  0099M  0111
RAK3UN R          001013  0076M  0080   0081   0082   0083   0084   0085
                          0086   0087   0089   0090   0092   0093
SDB    R ARGUMENT 000005  0001S  0054S  0058A  0064   0071   0110
SDBI   R          001015  0110M  0112   0114   0115
SRL    R          001017  0088M  0098
SRLB1  R          001021  0087M  0088
SRLB2  R          001023  0086M  0088
SRLT1  R          001025  0085M  0088
SRLT2  R          001027  0084M  0088
SRS    R          001031  0094M  0098
SRSB1  R          001033  0093M  0094
SRSB2  R          001035  0092M  0094
SRST1  R          001037  0090M  0094
SRST2  R          001041  0089M  0094
SUM    R          001043  0104M  0117   0119
SUMX   R          001045  0105M  0112M  0117   0118   0119
SUMX2  R          001047  0107M  0114M  0117   0118
SUMXY  R          001051  0108M  0115M  0118   0119
SUMY   R          001053  0106M  0113M  0118   0119
WI     R          001055  0111M  0113   0115
Y1     R          001057  0062M  0064
Y2     R          001061  0063M  0064
YM     R          001063  0064M  0075
Z3     R          001065  0069M  0071
Z4     R          001067  0070M  0071
ZM     R          001071  0071M  0075

_200              000627  0109   0116D

0000 ERRORS [<ESTCOF>FTN-REV15.3]
```

```
      SUBROUTINE FTPLOT(N,S,P,AK)
C
      DIMENSION S(1),P(1),AK(3)
C
C     OPEN PLOTER.
C
      CALL PLOTS(4,0,0)
C
C     PLOT DATA.
C
      CALL PTDATA(N,S,P)
C
C     PLOT FIT
C
      CALL PTQFIT(AK)
C
C     CLOSE PLOTTER.
C
      CALL PLOT(0.,0.,999)
C
      RETURN
      END
```

```
AK       R ARGUMENT 000006    0001S    0003S    0015A
FTPLOT   R          000000    0001S
N        I ARGUMENT 000003    0001S    0011A
P        R ARGUMENT 000005    0001S    0003S    0011A
PLOT     R EXTERNAL 000000    0019
PLOTS    R EXTERNAL 000000    0007
PTDATA   R EXTERNAL 000000    0011
PTQFIT   R EXTERNAL 000000    0015
S        R ARGUMENT 000004    0001S    0003S    0011A

0000 ERRORS [<FTPLOT>FTN-REV15.3]
```

SEQ NEST SUBROUTINE INSORT(Z,W,IF,IL)

```
      SUBROUTINE INSORT(Z,W,IF,IL)
C
C     (Z,W) ARE DATA PAIRS
C     SORTING W TO BE IN INCREASING ORDER
C     INSERTION SORTING USED
C     IF IS THE FIRST ELEMENT OF ARRAY TO BE SORTED
C     IL IS THE LAST ELEMENT OF ARRAY TO BE SORTED
C
      DIMENSION Z(1),W(1)
C
C     USING COMPILER TIME DOMAIN
C     COMPUTATION DONE IN PLACE
C
      ISTAR=IF+1
      DO 50 J=ISTAR,IL
C
C        SUBROUTINE TO INSERT W(IF+1) TO W(IL)
C
         CALL INSERT(Z,W,J,IF,IL)
50    CONTINUE
      RETURN
      END
```

```
SEQ NEST       SUBROUTINE INSERT(Z,W,I,NF,NL)

               SUBROUTINE INSERT(Z,W,I,NF,NL)

          C
          C    SUBROUTINE USED BY SUBROUTINE INSORT
          C    INSERTINQ W(I)
          C    (Z,W)ARE DATA PAIRS
          C    NF IS FIRST ELEMENT OF ARRAY ELIGIBLE FOR INSERTION
          C    NL IS LAST ELEMENT OF ARRAY ELIGIBLE FOR INSERTION
          C

               DIMENSION Z(1),W(1)

          C
          C    USING COMPILER TIME DOMAIN
          C    COMPUIATION DONE IN PLACE
          C

               J=I-1
               Y=W(I)
               X=Z(I)
          50   IF(Y .GE. W(J)) GO TO 100

          C
          C       MOVES W(J) ONE SPACE US AS Y WILL BE INSERTED TO
          C       THE LEFT OF W(J)
          C

                  W(J+1)=W(J)
                  Z(J+1)=Z(J)
                  ISTOP=NF-1
                  J=J-1
                  IF(J .EQ.  ISTOP) GO TO 100
                  GO TO 50
          100  W(J+1)=Y
               Z(J+1)=X
               RETURN
               END
IFTRAN STATISTICS
    53 CARDS READ
     0 ERROR(S) FOUND
1  SEQ NEST      SUBROUTINE MXINVR(A,B,IBAD)

                 SUBROUTINE MXINVR(A,B,IBAD)

          C  SUBROUTINE MXINVR FACTORS MATRIX A INTO LU CROUT DECOMPOSITITON
          C  AND COMPUTES ITS INVERSE, MATRIX B.

                 IMPLICIT DOUBLE PRECISION (A-H,O-Z)

          C

                 DIMENSION A(3,3),B(3,3)

          C

                 FOR(I=1 TO 3)
      1          .    FOR(J=1 TO 3)
      2          .    .    B(I,J)=0.
      2          .    .    IF(I.EQ.J) B(I,J)=1.
      1          .    END FOR
                 END FOR

          C

                 IBAD=0
                 FOR(K=1 TO 3)
      1          .    FOR(I=K TO 3)
      2          .    .    DINN1=0.DO
      2          .    .    FOR(L=1 TO (K-1))
```

```
 3               .    .    .    DINN1=DINN1+A(I,L)*A(L,K)
 2               .    .    END FOR
 2               .    .    A(I,K)=A(I,K)-DINN1
 1               .    END FOR

 1      C        *    TEST FOR SINGULARITY.

 1               .    IF(A(K,K).NE.0)
 2               .    .    FOR(J=(K+1) TO 3)
 3               .    .    .    DINN2=0.DO
 3               .    .    .    FOR(L=1 TO (K-1))
 4               .    .    .    .    DINN2=DINN2+A(K,L)*A(L,J)
 3               .    .    .    END FOR
 3               .    .    .    A(K,J)=(A(K,J)-DINN2)/A(K,K)
 2               .    .    END FOR
 1               .    ELSE
 2               .    .    IBAD=1
 2               .    .    K=4
 1               .    END IF
                 END FOR

        C

                 IF(IBAD.NE.1)
 1               .    FOR(M=1 TO 3)
 2               .    .    FOR(I=1 TO 3)
 3               .    .    .    DINN1=0.DO
 3               .    .    .    FOR(K=1 TO (I-1))
 4               .    .    .    .    DINN1=DINN1+A(I,K)*B(K,M)
1  SEQ NEST      SUBROUTINE MXINVR(A,B,IBAD)
 3               .    .    .    END FOR
 3               .    .    .    B(I,M)=(B(I,M)-DINN1)/A(I,I)
 2               .    .    END FOR
 2               .    .    FOR(I=3 TO 1 STEP -1)
 3               .    .    .    DINN1=0.DO
 3               .    .    .    FOR(J=(I+1) TO 3)
 4               .    .    .    .    DINN1=DINN1+A(I,J)*B(J,M)
 3               .    .    .    END FOR
 3               .    .    .    B(I,M)=B(I,M)-DINN1
 2               .    .    END FOR
 1               .    END FOR
                 ELSE
 1               .    RETURN
                 END IF

        C

                 RETURN
                 END

IFTRAN STATISTICS
    61 CARDS READ
     0 ERROR(S) FOUND
```

```
      SUBROUTINE PDFIT(N,SDBS,PS,THS)
C
C
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      INTEGER FILE6(3),CODE
      REAL CASES,PFAS,THS(3),SNRS(12),PCS(12),PDFS(12),
     1  ERRORS(12),ERRMS,SDERRS,SEC1,SEC2,SECS,SES,SDBS(12),PS(12)
C
      DIMENSION PD(12),ED(12),WK(3),ISTOP(12),ST(12),ERROR(12)
      DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
      DIMENSION IP(3),STP(3)
C
      COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,KS,JV
C
      DATA WK/0.1D0,0.1D0,0.1D0/
      DATA ST/0.001D0,0.001D0,0.001D0,0.001D0,0.001D0,0.001D0,
     10.001D0,0.001D0,0.001D0,0.001D0,0.001D0,0.001D0/
C
C  INITIALIZATION.
C
      DO 10 I=1,N
          PD(I)=DBLE(PS(I))
          SNR(I)=DBLE(10.**(SDBS(I)*.1))
 10   CONTINUE
      TH(1)=1.D0/DBLE(THS(1))
      TH(2)=DBLE(THS(2))
      TH(3)=DBLE(THS(3))
      L=5
      M=0
      IP(1)=0
      KS=0
      IP(2)=0
      IP(3)=0
      ISS=0
      DEL(1)=0.0001D0
      DEL(2)=0.0001D0
      DEL(3)=0.0001D0
      STP(1)=0.000001D0
      STP(2)=0.000001D0
      STP(3)=0.000001D0
      ISUM=0
      DMIN=1.0D-2
      WEIGH=1.0D0
      SK=1.0D0
      DO 20 I=1,12
      ISTOP(I)=0
 20   CONTINUE
      JV=N
C
C  COMPUTE INITIAL DECUPLING MATRIX.
      CALL VBAR
C
      CALL DCPLE
      KSMAX=500
      DO 160 KS=1,KSMAX
      M=M+1
      CALL VBAR
C
C  COMPUTE DECUPLING MATRIX.
C
      IF(M.GT.L) CALL DCPLE
C
C  FORM ERRORS.
C
      DO 110 I=1,JV
      ED(I)=(PDF(I)-PD(I))*WEIGH
      IF(DABS(ED(I)).LE.ST(I))ISTOP(I)=1
      ISUM=ISUM+ISTOP(I)
 110  CONTINUE
```

```
C
C   COMPUTE WEIGHTED ERRORS.
C
        DO 130 I=1,3
        DINC(I)=0.0D0
        DO 120 K=1,JV
        DINC(I)=DINC(I)+W(I,K)*ED(K)
120     CONTINUE
C
C   COMPUTE UPDATED PARAMETER ESTIMATE.
C
        TH(I)=TH(I)-WK(I)*DINC(I)/SK
        SS=WK(I)*DINC(I)
        IF(DABS(SS).LE.STP(I))IP(I)=1
        ISS=ISS+IP(I)
C
C   CHECK IF ESTIMATE HAS CONVERGED. IF YES, SET ISTOP(I)
C   EQUAL TO 1.
C
130     CONTINUE
C
C   HAVE ALL THREE ESTIMATES CONVERGED?
C
        IF(ISUM.EQ.JV) GO TO 170
        IF(ISS.EQ.3) GO TO 170
        DO 140 I=1,JV
        ISTOP(I)=0
140     CONTINUE
        DO 150 I=1,3
        IP(I)=0
150     CONTINUE
        ISS=0
        ISUM=0
160     CONTINUE
170     CONTINUE
        IF(ISUM .EQ. JV)WRITE(1,1001)
        IF(ISS .EQ. 3)WRITE(1,1002)
        IF(KS .GE. KSMAX)WRITE(1,1003)
        S2=0.0D0
        SS=0.0D0
        DO 180 I=1,JV
        ERROR(I)=PD(I)-PDF(I)
        S2=S2+ERROR(I)**2
        SS=SS+ERROR(I)
180     CONTINUE
        ERRM=SS/DBLE(FLOAT(JV))
        SDERR=DSQRT(S2/DBLE(FLOAT(JV))-ERRM*ERRM)
C
C   DOUBLE PRECISION TO SINGLE PRECISION CONVERSION.
C
        CASES=SNGL(CASE)
        PFAS=SNGL(PFA)
        DO 190 I=1,3
        THS(I)=SNGL(TH(I))
190     CONTINUE
        THS(1)=1./THS(1)
        DO 200 I=1,JV
        SNRS(I)=SNGL(SNR(I))
        PDS(I)=SNGL(PD(I))
        PDFS(I)=SNGL(PDF(I))
        ERRORS(I)=SNGL(ERROR(I))
200     CONTINUE
        ERRMS=SNGL(ERRM)
        SDERRS=SNGL(SDERR)
C
C   WRITE TO OUTPUT FILE.
C
        CALL PDFOUT(N,THS,SNRS,PDS,PDFS,ERRORS,ERRMS,SDERRS)
C
        RETURN
1001    FORMAT('PDFIT: TERMINATION 1, SMALL ERRORS')
1002    FORMAT('PDFIT: TERMINATION 2, SMALL COF. CHANGE')
```

```
(0145)   1003  FORMAT('PDFIT: TERMINATION 3, MAX. ITERATIONS')
(0146)         END
```

CASE	D		001600	0123A						
CASES	R		001604	0009S	0123M					
CODE	I		000000	0008S						
DABS	D	EXTERNAL	000000	0070	0086					
DBLE	D	EXTERNAL	000000	0025	0026	0028	0029	0030	0118	0119
DCPLE	D	EXTERNAL	000000	0056	0064					
DEL	D	/COM/	000154	0013S	0016S	0038M	0039M	0040M		
DINC	D	/COM/	000170	0013S	0016S	0077M	0079M	0084	0085	
DMIN	D		001616	0045M						
DSQRT	D	EXTERNAL	000000	0119						
ED	D		000104	0012S	0069M	0070A	0079			
ERRM	D		001622	0118M	0119	0135A				
ERRMS	R		001626	0009S	0135M	0140A				
ERROR	D		000164	0012S	0114M	0115	0116	0133A		
ERRORS	R		000244	0009S	0133M	0140A				
FILE6	I		000274	0008S						
FLOAT	D	EXTERNAL	000000	0118	0119					
I	I		001630	0024M	0025	0026	0048M	0049	0068M	0069
				0070	0071	0076M	0077	0079	0084	0085
				0086	0087	0098M	0099	0101M	0102	0113M
				0114	0115	0116	0125M	0126	0129M	0130
				0131	0132	0133				
IP	I		000277	0014S	0033M	0035M	0036M	0086M	0087	0102M
ISS	I		001633	0037M	0087M	0097	0104M	0109		
ISTOP	I		000302	0012S	0049M	0070M	0071	0099M		
ISUM	I		001634	0044M	0071M	0096	0105M	0108		
JV	I	/COM/	000426	0016S	0051M	0068	0078	0096	0098	0108
				0113	0118	0119	0129			
K	I		001635	0078M	0079					
KS	I	/COM/	000425	0016S	0034M	0058M	0110			
KSMAX	I		001636	0057M	0058	0110				
L	I		001637	0031M	0064					
M	I	/COM/	000424	0016S	0032M	0059M	0064			
N	I	ARGUMENT	000003	0001S	0024	0051	0140A			
PD	D		000316	0012S	0025M	0069	0114	0131A		
PDF	D	/COM/	000060	0013S	0016S	0069	0114	0132A		
PDFIT	D		000000	0001S						
PDFOUT	D	EXTERNAL	000000	0140						
PDFS	R		000376	0009S	0132M	0140A				
PDS	R		000426	0009S	0131M	0140A				
PFA	D		001640	0124A						
PFAS	R		001644	0009S	0124M					
PS	R	ARGUMENT	000005	0001S	0009S	0025A				
S2	D		001650	0111M	0115M	0119				
SDBS	R	ARGUMENT	000004	0001S	0009S	0026				
SDERR	D		001654	0119M	0136A					
SDERRS	R		001660	0009S	0136M	0140A				
SEC1	R		000000	0009S						
SEC2	R		000000	0009S						
SECS	R		000000	0009S						
SES	R		000000	0009S						
SK	D		001662	0047M	0084					
SNGL	R	EXTERNAL	000000	0123	0124	0126	0130	0131	0132	0133
				0135	0136					
SNR	D	/COM/	000000	0013S	0016S	0026M	0130A			
SNRS	R		000456	0009S	0130M	0140A				
SS	D		001666	0085M	0086A	0112M	0116M	0118		
ST	D		00[illegible]24	0012S	0019I	0070				
STP	D		000506	0014S	0041M	0042M	0043M	0086		
TH	D	/COM/	000140	0013S	0016S	0028M	0029M	0030M	0084M	0126A
THS	R	ARGUMENT	000006	0001S	0009S	0028A	0029A	0030A	0126M	0128M
				0140A						
VBAR	D	EXTERNAL	000000	0054	0060					
W	D	/COM/	000204	0013S	0016S	0079				
WEIGH	D		001672	0046M	0069					
WK	D		000010	0012S	0018I	0084	0085			
_10			000565	0024	00[illegible]D					
_1001			001466	0108	01[illegible]					
_1002			001512	0109	014[illegible]					
_1003			001541	0110	0143D					
_110			000764	0043	0072D					

```
_120             001032   0078   0080D
_130             001103   0076   0092D
_140             001126   0098   0100D
_150             001141   0101   0103D
_160             001152   0058   0106D
_170             001160   0096   0097    0107D
_180             001244   0113   0117D
_190             001346   0125   0127D
_20              000700   0048   0050D
_200             001430   0129   0134D

0000 ERRORS [<PDFIT >FTN-REV15.3]
```

```
      SUBROUTINE PDFOUT(N,TH,SNR,PD,PDF,ERROR,ERRM,SDERR)
C
C SYSCOM>KEYS.F      MNEMONIC KEYS FOR FILE SYSTEM (FTN)      31 MAY, 197
      NOLIST
C SYSCOM>ERRD.F      MNEMONIC CODES FOR FILE SYSTEM (FTN)      6 SEPT, 19
      NOLIST
      INTEGER FILE6(6),CODE
      DIMENSION TH(3),SNR(1),PD(1),PDF(1),ERROR(1)
C
C  ENTER  OUTPUT FILE NAME.
C
10    CONTINUE
      WRITE(1,1200)
      READ(1,1300)FILE6
C
C     SEARCH TO SEE IF FILE ALREADY EXISTS.
C
      CALL SRCH$$(K$EXST,FILE6,12,2,0,ICODE)
      IF(ICODE .EQ. E$FNTF)GO TO 20
         WRITE(1,1006)
         READ(1,*)IOK
         IF(IOK .EQ. 1)GO TO 20
         GO TO 10
20    CONTINUE
C
C  OPEN  OUTPUT FILE.
C
      CALL SRCH$$(K$WRIT,FILE6,12,2,0,CODE)
      IF(CODE.NE.0) CALL ERRPR$(K$NRTN,CODE,0,0,'PFDOUT',6)
C
C
      WRITE(1,6666)
C
6666  FORMAT('ENTER: CASE PFA NP NB'/)
      READ(1,*)CASE,PFA,NP,NB
C
      GO TO (210,220,230,240), NB
210   WRITE(6,1431)
      GO TO 250
220   WRITE(6,1432)
      GO TO 250
230   WRITE(6,1433)
      GO TO 250
240   WRITE(6,1434)
C
250   WRITE(6,1440)
      WRITE(6,1460)
      WRITE(6,1500)
      WRITE(6,1600)
      WRITE(6,1700)CASE,PFA,NP
      WRITE(6,1800)
      WRITE(6,1900)
      WRITE(6,2000)
      WRITE(6,2100)
      WRITE(6,2200)TH(1),TH(2),TH(3)
      WRITE(6,2300)
      WRITE(6,2400)
      WRITE(6,2500)
      WRITE(6,2600)
      WRITE(6,2700)(I,SNR(I),PD(I),PDF(I),ERROR(I),I=1,N)
      WRITE(6,2800)ERRM,SDERR
C
C  REWIND OUTPUT FILE.
C
      REWIND 6
C
C  CLOSE  OUTPUT FILE.
C
      CALL SRCH$$(K$CLOS,0,0,2,0,CODE)
      IF(CODE.EQ.0) GO TO 300
      CALL ERRPR$(K$NRTN,CODE,0,0,'FDFOUT',6)
300   CONTINUE
```

```
      RETURN
C
C  FORMATS.
C
1006  FORMAT('PDFOUT:  * EXISTING FILE * IF OK TO MOD ENTER 1, ELSE 0')
1200  FORMAT(///'ENTER OUTPUT FILE NAME - 12')
1300  FORMAT(6A2)
1431  FORMAT(////36X, 'RAYLEIGH')
1432  FORMAT(////36X, 'LOG - NORMAL -3DB')
1433  FORMAT(////36X, 'LOG - NORMAL - 6DB')
1434  FORMAT(////36X, 'WIEBULL - .33')
1440  FORMAT(////36X, 'INPUT PARAMETERS')
1460  FORMAT(36X, '----- ----------')
1500  FORMAT(//15X, 'SWERLING CASE NO. ', 10X, 'PFA', 12X,
     1   'PULSES INTEGRATED')
1600  FORMAT(15X, '-------- ---- ---', 10X, '---', 12X, '------ ----------')
1700  FORMAT(//22X,F3.1,11X,E11.1,17X,I3)
1800  FORMAT(////34X, 'PARAMETER ESTIMATES')
1900  FORMAT(34X, '--------- ---------')
2000  FORMAT(//18X, 'PARAMETER 1', 10X, 'PARAMETER 2', 10X, 'PARAMETER 3')
2100  FORMAT(18X, '--------- -', 10X, '--------- -', 10X, '--------- -')
2200  FORMAT(//14X,F14.7,7X,F14.7,7X,F14.7)
2300  FORMAT(////42X, 'FIT')
2400  FORMAT(42X, '---')
2500  FORMAT(//18X, 'N', 10X, 'SNR', 10X, 'PD', 10X, 'PDF', 10X, 'ERROR')
2600  FORMAT(18X, '-', 10X, '---', 10X, '--', 10X, '---', 10X, '-----')
2700  FORMAT(17X,I2,6X,E11.4,3X,F8.5,4X,F8.5,6X,F8.5)
2800  FORMAT(///19X'MEAN ERROR = ',F7.5,12X,'RMS ERROR = ',F7.5)
C
      END
```

```
CASE    R               001357   0033M  0048
CODE    I               001361   0005S  0026A  0027A  0067A  0068  0069A
E$BCOD  I  PARAMETER             0004S
E$BDAM  I  PARAMETER             0004S
E$BFSV  I  PARAMETER             0004S
E$BFTS  I  PARAMETER             0004S
E$BKEY  I  PARAMETER             0004S
E$BNAM  I  PARAMETER             0004S
E$BOF   I  PARAMETER             0004S
E$BPAR  I  PARAMETER             0004S
E$BPAS  I  PARAMETER             0004S
E$BSGN  I  PARAMETER             0004S
E$BSUN  I  PARAMETER             0004S
E$BTRN  I  PARAMETER             0004S
E$BUFD  I  PARAMETER             0004S
E$BUNT  I  PARAMETER             0004S
E$DIRE  I  PARAMETER             0004S
E$DISK  I  PARAMETER             0004S
E$DKFL  I  PARAMETER             0004S
E$DNS   I  PARAMETER             0004S
E$DNTE  I  PARAMETER             0004S
E$DVIU  I  PARAMETER             0004S
E$EOF   I  PARAMETER             0004S
E$EXST  I  PARAMETER             0004S
E$FABT  I  PARAMETER             0004S
E$FBST  I  PARAMETER             0004S
E$FDEL  I  PARAMETER             0004S
E$FDFL  I  PARAMETER             0004S
E$FIFC  I  PARAMETER             0004S
E$FITB  I  PARAMETER             0004S
E$FIUS  I  PARAMETER             0004S
E$FNTF  I  PARAMETER             0004S  0017
E$FNTS  I  PARAMETER             0004S
E$FONC  I  PARAMETER             0004S
E$FUIU  I  PARAMETER             0004S
E$IREM  I  PARAMETER             0004S
E$NASS  I  PARAMETER             0004S
E$NATT  I  PARAMETER             0004S
E$NMLG  I  PARAMETER             0004S
E$NPHA  I  PARAMETER             0004S
E$NRIT  I  PARAMETER             0004S
E$NTIM  I  PARAMETER             0004S
E$NTSD  I  PARAMETER             0004S
E$NTUD  I  PARAMETER             0004S
E$NULL  I  PARAMETER             0004S
E$OLDP  I  PARAMETER             0004S
E$PTRM  I  PARAMETER             0004S
E$RLDN  I  PARAMETER             0004S
E$ROOM  I  PARAMETER             0004S
E$SDER  I  PARAMETER             0004S
E$SEMO  I  PARAMETER             0004S
E$SHUT  I  PARAMETER             0004S
E$SUNO  I  PARAMETER             0004S
E$TMRU  I  PARAMETER             0004S
E$TMUL  I  PARAMETER             0004S
E$UIUS  I  PARAMETER             0004S
E$UNOP  I  PARAMETER             0004S
ERRM    R  ARGUMENT     000011   0001S  0059
ERROR   R  ARGUMENT     000010   0001S  0006S  0058
ERRPR$  R  EXTERNAL     000000   0027   0069
FILE6   I               000014   0005S  0012M  0016A  0026A
I       I               001362   0058M
ICODE   I               001363   0016A  0017
IOK     I               001364   0019M  0020
K$ALLD  I  PARAMETER             0003S
K$CACC  I  PARAMETER             0003S
K$CLOS  I  PARAMETER             0003S  0067
K$CONV  I  PARAMETER             0003S
K$CURR  I  PARAMETER             0003S
K$DELE  I  PARAMETER             0003S
K$DMPB  I  PARAMETER             0003S
K$DTIM  I  PARAMETER             0003S
```

K$ENTR	I		000000	0003S		
K$EXST	I	PARAMETER		0003S	0016	
K$FREE	I	PARAMETER		0003S		
K$FULL	I	PARAMETER		0003S		
K$GOND	I	PARAMETER		0003S		
K$GPOS	I	PARAMETER		0003S		
K$HOME	I	PARAMETER		0003S		
K$ICUR	I	PARAMETER		0003S		
K$IMFD	I	PARAMETER		0003S		
K$IRTN	I	PARAMETER		0003S		
K$ISEG	I	PARAMETER		0003S		
K$IUFD	I	PARAMETER		0003S		
K$MENT	I		000000	0003S		
K$MSIZ	I	PARAMETER		0003S		
K$MVNT	I	PARAMETER		0003S		
K$NAME	I	PARAMETER		0003S		
K$NDAM	I	PARAMETER		0003S		
K$NRTN	I	PARAMETER		0003S	0027	0069
K$NSAM	I	PARAMETER		0003S		
K$NSGD	I	PARAMETER		0003S		
K$NSGS	I	PARAMETER		0003S		
K$POSA	I	PARAMETER		0003S		
K$POSN	I	PARAMETER		0003S		
K$POSR	I	PARAMETER		0003S		
K$PREA	I	PARAMETER		0003S		
K$PRER	I	PARAMETER		0003S		
K$PROT	I	PARAMETER		0003S		
K$RDWR	I	PARAMETER		0003S		
K$READ	I	PARAMETER		0003S		
K$RPOS	I	PARAMETER		0003S		
K$RSUB	I	PARAMETER		0003S		
K$RWLK	I	PARAMETER		0003S		
K$SENT	I		000000	0003S		
K$SETC	I	PARAMETER		0003S		
K$SETH	I	PARAMETER		0003S		
K$SPOS	I	PARAMETER		0003S		
K$SRTN	I	PARAMETER		0003S		
K$TRNC	I	PARAMETER		0003S		
K$UPOS	I	PARAMETER		0003S		
K$WRIT	I	PARAMETER		0003S	0026	
N	I	ARGUMENT	000003	0001S	0058	
NB	I		001365	0033M	0035	
NP	I		001366	0033M	0048	
PD	R	ARGUMENT	000006	0001S	0006S	0058
PDF	R	ARGUMENT	000007	0001S	0006S	0058
PDFOUT	R		000000	0001S		
PFA	R		001367	0033M	0048	
SDERR	R	ARGUMENT	000012	0001S	0059	
SNR	R	ARGUMENT	000005	0001S	0006S	0058
SRCH$$	R	EXTERNAL	000000	0016	0026	0067
TH	R	ARGUMENT	000004	0001S	0006S	0053
_10			000022	0010D	0021	
_1006			000457	0018	0075D	
_1200			000515	0011	0076D	
_1300			000537	0012	0077D	
_1431			000543	0036	0078D	
_1432			000556	0038	0079D	
_1433			000576	0040	0080D	
_1434			000616	0042	0081D	
_1440			000634	0044	0082D	
_1460			000653	0045	0083D	
_1500			000670	0046	0084D	
_1600			000730	0047	0086D	
_1700			000767	0048	0087D	
_1800			001007	0049	0088D	
_1900			001030	0050	0089D	
_20			000071	0017	0020	0022D
_2000			001047	0051	0090D	
_210			000165	0035	0036D	
_2100			001105	0052	0091D	
_220			000172	0035	0038D	

```
2200                 001142   0053   0092D
230                  000177   0035   0040D
2300                 001163   0054   0093D
240                  000204   0035   0042D
2400                 001174   0055   0094D
250                  000210   0037   0039    0041    0044D
2500                 001203   0056   0095D
2600                 001236   0057   0096D
2700                 001270   0058   0097D
2800                 001316   0059   0098D
300                  000455   0068   0070D
6666                 000120   0030   0032D

0000 ERRORS [<PDFOUT>FTN-REV15.3]
```

```
      SUBROUTINE PTDATA(N,X,Y)
      DIMENSION X(1),Y(1),LABLEX(16),LABLEY(16)
      COMMON /PTCOM/XORIG,YORIG,XMAX,XMIN,YMIN,SCALEX,SCALEY,
     $SIZEX
C
C     FIND MAX & MIN VALUES OF X & Y.
C
      XMIN=1.E32
      YMIN=1.E32
      XMAX=-XMIN
      YMAX=-YMIN
      DO 10 I=1,N
           IF(X(I) .GT.  XMAX)XMAX=X(I)
           IF(X(I) .LT.  XMIN)XMIN=X(I)
           IF(Y(I) .GT.  YMAX)YMAX=Y(I)
           IF(Y(I) .LT.  YMIN)YMIN=Y(I)
10    CONTINUE
C
C     INPUT PLOT SIZE.
C
      WRITE(1,1001)
      READ(1,*)SIZEX,SIZEY
C
C     RE-ADJUST XMAX-XMIN,YMAX-YMIN TO MAKE A NICE PLOT
C
      CALL SCALE(XMAX,XMIN,DELX,ANTICX)
      CALL SCALE(YMAX,YMIN,DELY,ANTICY)
C
C     COMPUTE SCALE FACTORS
C
      SCALEX=(XMAX-XMIN)/SIZEX
      SCALEY=(YMAX-YMIN)/SIZEY
      H=.022*SIZEX
C
C     SET ORIGIN
C
      XORIG=2.0
      YORIG=1.5
      CALL PLOT(XORIG,YORIG,-3)
C
C     DRAW GRID & LABEL.
C
      NTICX=IFIX(ANTICX)
      NTICY=IFIX(ANTICY)
      CALL BORDER(SIZEX,SIZEY,NTICX,NTICY,1,1,H)
      WRITE(1,1002)
      READ(1,1003)LABLEX
      WRITE(1,1004)
      READ(1,1003)LABLEY
      CALL AXLABL(SIZEX,SIZEY,H,XMIN,DELX,XMAX,1,1,YMIN,DELY,
     $YMAX,1,1,LABLEX,32,LABLEY,32)
C
C     PLOT DATA
C
      DO 100 I=1,N
           XP=(X(I)-XMIN)/SCALEX
           YP=(Y(I)-YMIN)/SCALEY
           CALL PLOT(XP,YP,3)
           H2=H
           CALL CSYMBL(XP,YP,H2,2,0.,-2)
100   CONTINUE
C
C     RE-SET ORIGIN.
C
      CALL PLOT(-XORIG,-YORIG,-3)
C
1001  FORMAT('ENTER XDIM(IN),YDIM(IN)')
1002  FORMAT('ENTER X AXIS LABEL, FORMAT 32A1')
1003  FORMAT(32A1)
1004  FORMAT('ENTER YAXIS LABEL, FORMAT 32A1')
C
      RETURN

      END
```

```
ANTICX  R            000533   0026A   0043
ANTICY  R            000535   0027A   0044
AXLABL  R  EXTERNAL  000000   0050
BORDER  R  EXTERNAL  000000   0045
CSYMBL  R  EXTERNAL  000000   0060
DELX    R            000537   0026A   0050A
DELY    R            000541   0027A   0050A
H       R            000543   0033M   0045A   0050A   0059
H2      R            000545   0059M   0060A
I       I            000547   0012M   0013    0014    0015    0016    0055M   0056
                              0057
IFIX    I  EXTERNAL  000000   0043    0044
LABLEX  I            000007   0002S   0047M   0050A
LABLEY  I            000027   0002S   0049M   0050A
N       I  ARGUMENT  000003   0001S   0012    0055
NTICX   I            000552   0043M   0045A
NTICY   I            000553   0044M   0045A
PLOT    R  EXTERNAL  000000   0039    0058    0065
PTDATA  R            000000   0001S
SCALE   R  EXTERNAL  000000   0026    0027
SCALEX  R  /PTCOM/   000012   0003S   0031M   0056
SCALEY  R  /PTCOM/   000014   0003S   0032M   0057
SIZEX   R  /PTCOM/   000016   0003S   0022M   0031    0033    0045A   0050A
SIZEY   R            000560   0022M   0032    0045A   0050A
X       R  ARGUMENT  000004   0001S   0002S   0013    0014    0056
XMAX    R  /PTCOM/   000004   0003S   0010M   0013M   0026A   0031    0050A
XMIN    R  /PTCOM/   000006   0003S   0008M   0010    0014M   0026A   0031    0050A
                              0056
XORIG   R  /PTCOM/   000000   0003S   0037M   0039A   0065
XP      R            000562   0056M   0058A   0060A
Y       R  ARGUMENT  000005   0001S   0002S   0015    0016    0057
YMAX    R            000564   0011M   0015M   0027A   0032    0050A
YMIN    R  /PTCOM/   000010   0003S   0009M   0011    0016M   0027A   0032    0050A
                              0057
YORIG   R  /PTCOM/   000002   0003S   0038M   0039A   0065
YP      R            000566   0057M   0058A   0060A

_10                  000144   0012    0017D
_100                 000413   0055    0061D
_1001                000441   0021    0067D
_1002                000460   0046    0068D
_1003                000503   0047    0049    0069D
_1004                000507   0048    0070D

0000 ERRORS [<PTDATA>FTN-REV15.3]
```

```
      SUBROUTINE PTQFIT(AK)
C
      DIMENSION AK(3)
      COMMON /PTCOM/XORIG,YORIG,XMAX,XMIN,YMIN,SCALEX,SCALEY,
     *SIZEX
C
      PF(A,B,C,S)=1.-(1.+(A*S)**B)**(-C)
C
C     SET ORIGIN.
C
      CALL PLOT(XORIG,YORIG,-3)
C
C     COMPUTE VALUES & PLOT.
C
      J=1
      X=XMAX
      DELX=SIZEX/100.
 10   IF(X .LT. XMIN)GO TO 100
         Y=PF(AK(1),AK(2),AK(3),X)
         XP=(X-XMIN)/SCALEX
         YP=(Y-YMIN)/SCALEY
         IF(J .EQ. 1)CALL PLOT(XP,YP,3)
         CALL PLOT(XP,YP,2)
         J=2
         X=X-DELX
      GO TO 10
 100  CONTINUE
C
C     RESET ORIGIN.
C
      CALL PLOT(-XORIG,-YORIG,-3)
C
      RETURN
      END
```

```
A        R           000000   0007
AK       R ARGUMENT  000003   0001S   0003S   0019A
B        R           000000   0007
C        R           000000   0007
DELX     R           000201   0017M   0025
J        I           000203   0015M   0022    0024M
PF       R           000005   0007S   0019
PLOT     R EXTERNAL  000000   0011    0022    0023    0031
PTQFIT   R           000000   0001S
S        R           000000   0007
SCALEX   R /PTCOM/   000012   0004S   0020
SCALEY   R /PTCOM/   000014   0004S   0021
SIZEX    R /PTCOM/   000016   0004S   0017
X        R           000212   0016M   0018    0019A   0020    0025M
XMAX     R /PTCOM/   000004   0004S   0016
XMIN     R /PTCOM/   000006   0004S   0018    0020
XORIG    R /PTCOM/   000000   0004S   0011A   0031
XP       R           000214   0020M   0022A   0023A
Y        R           000216   0019M   0021
YMIN     R /PTCOM/   000010   0004S   0021
YORIG    R /PTCOM/   000002   0004S   0011A   0031
YP       R           000220   0021M   0022A   0023A

_10                  000062   0018D   0026
_100                 000157   0018    0027D

0000 ERRORS [<PTQFIT>FTN-REV15.3]
```

```
      SUBROUTINE SCALE(RMAX,RMIN,RINC,ANTIC)
      RMIN=0.
      IF(RMAX .GT. 1.)GO TO 5
         RMAX=1.
         GO TO 100
5     IF(RMAX .GT. 10.)GO TO 10
         RMAX=10.
         GO TO 100
10    IF(RMAX .GT. 20.)GO TO 20
         RMAX=20.
         GO TO 100
20    IF(RMAX .GT. 50.)GO TO 30
         RMAX=50.
         GO TO 100
30    IF(RMAX .GT. 100.)GO TO 40
         RMAX=100.
         GO TO 100
40    IF(RMAX .GT. 150.)GO TO 50
         RMAX=150.
         GO TO 100
50    IF(RMAX .GT. 200.)GO TO 60
         RMAX=200.
         GO TO 100
60    IF(RMAX .GT. 250.)GO TO 70
         RMAX=250.
         GO TO 100
70    IF(RMAX .GT. 300.)GO TO 80
         RMAX=300.
         GO TO 100
80    RMAX=500.
100   CONTINUE
      RINC=(RMAX-RMIN)/10.
      ANTIC=10.
      RETURN
      END
```

```
ANTIC    R ARGUMENT 000006    0001S   0033M
RINC     R ARGUMENT 000005    0001S   0032M
RMAX     R ARGUMENT 000003    0001S   0003    0004M   0006    0007M   0009    0010M
                              0012    0013M   0015    0016M   0018    0019M   0021
                              0022M   0024    0025M   0027    0028M   0030M   0032
RMIN     R ARGUMENT 000004    0001S   0002M   0032
SCALE    R          000000    0001S

_10                 000037    0006    0009D
_100                000151    0005    0008    0011    0014    0017    0020    0023
                              0026    0029    0031D
_20                 000051    0009    0012D
_30                 000063    0012    0015D
_40                 000075    0015    0018D
_5                  000025    0003    0006D
_50                 000107    0018    0021D
_60                 000121    0021    0024D
_70                 000133    0024    0027D
_80                 000145    0027    0030D

0000 ERRORS [<SCALE >FTN-REV15.3]
```

```
      SUBROUTINE QFIN(N,SDB,P)
C
C SYSCOM>KEYS.F      MNEMONIC KEYS FOR FILE SYSTEM (FTN)      31 MAY
      NOLIST
      DIMENSION SDB(1),P(1)
      DIMENSION NAMEF(6)
C
C     READ INPUT FILE NAME.
C
      WRITE(1,1001)
      READ(1,1002)NAMEF
C
C     OPEN FILE FOR READING.
C
      CALL SRCH$$(K$READ,NAMEF,12,1,0,ICODE)
      IF(ICODE .NE. 0)CALL ERRPR$(K$NRTN,ICODE,0,0,0,0)
C
C     READ SDB,P VALUES.
C
      READ(5,*)N
      IF(N .GT. 50)WRITE(1,1003)
      DO 50 I=1,N
         READ(5,*)SDB(I),P(I)
50    CONTINUE
C
C     CLOSE INPUT FILE.
C
      CALL SRCH$$(K$CLOS,NAMEF,12,1,0,ICODE)
      IF(ICODE .NE. 0)CALL ERRPR$(K$NRTN,ICODE,0,0,0,0)
C
      RETURN
C
1001  FORMAT('ENTER INPUT FILE NAME,FORMAT 6A2')
1002  FORMAT(6A2)
1003  FORMAT('INPUT: **** ERROR, TOO MANY DATA PAIRS FOR STORAGE **
C
      END
```

```
ERRPR$  R  EXTERNAL  000000   0015    0028
I       I            000244   0021M   0022
ICODE   I            000245   0014A   0015A   0027A   0028A
K$ALLD  I  PARAMETER          0003S
K$CACC  I  PARAMETER          0003S
K$CLOS  I  PARAMETER          0003S   0027
K$CONV  I  PARAMETER          0003S
K$CURR  I  PARAMETER          0003S
K$DELE  I  PARAMETER          0003S
K$DMPB  I  PARAMETER          0003S
K$DTIM  I  PARAMETER          0003S
K$ENTR  I            000000   0003S
K$EXST  I  PARAMETER          0003S
K$FREE  I  PARAMETER          0003S
K$FULL  I  PARAMETER          0003S
K$GOND  I  PARAMETER          0003S
K$GPOS  I  PARAMETER          0003S
K$HOME  I  PARAMETER          0003S
K$ICUR  I  PARAMETER          0003S
K$IMFD  I  PARAMETER          0003S
K$IRTN  I  PARAMETER          0003S
K$ISEG  I  PARAMETER          0003S
K$IUFD  I  PARAMETER          0003S
K$MENT  I            000000   0003S
K$MSIZ  I  PARAMETER          0003S
K$MVNT  I  PARAMETER          0003S
K$NAME  I  PARAMETER          0003S
K$NDAM  I  PARAMETER          0003S
K$NRTN  I  PARAMETER          0003S   0015    0028
K$NSAM  I  PARAMETER          0003S
K$NSGD  I  PARAMETER          0003S
K$NSGS  I  PARAMETER          0003S
K$POSA  I  PARAMETER          0003S
K$POSN  I  PARAMETER          0003S
K$POSR  I  PARAMETER          0003S
K$PREA  I  PARAMETER          0003S
K$PRER  I  PARAMETER          0003S
K$PROT  I  PARAMETER          0003S
K$RDWR  I  PARAMETER          0003S
K$READ  I  PARAMETER          0003S   0014
K$RPOS  I  PARAMETER          0003S
K$RSUB  I  PARAMETER          0003S
K$RWLK  I  PARAMETER          0003S
K$SENT  I            000000   0003S
K$SETC  I  PARAMETER          0003S
K$SETH  I  PARAMETER          0003S
K$SPOS  I  PARAMETER          0003S
K$SRTN  I  PARAMETER          0003S
K$TRNC  I  PARAMETER          0003S
K$UPOS  I  PARAMETER          0003S
K$WRIT  I  PARAMETER          0003S
N       I  ARGUMENT  000003   0001S   0019M   0020    0021
NAMEF   I            000007   0005S   0010M   0014A   0027A
P       R  ARGUMENT  000005   0001S   0004S   0022M
QFIN    R            000000   0001S
SDB     R  ARGUMENT  000004   0001S   0004S   0022M
SRCH$$  R  EXTERNAL  000000   0014    0027

_1001                000154   0009    0032D
_1002                000177   0010    0033D
_1003                000203   0020    0034D
_50                  000122   0021    0023D

0000 ERRORS [<QFIN    >FTN-REV15.3]
```

```
      SUBROUTINE QFOUT(SDB,S,P,PFIT,ERR,N,AK)
C
C SYSCOM>KEYS.F      MNEMONIC KEYS FOR FILE SYSTEM (FTN)      31 MAY, 1977
      NOLIST
C SYSCOM>ERRD.F      MNEMONIC CODES FOR FILE SYSTEM (FTN)      6 SEPT, 1977
      NOLIST
C
      DIMENSION SDB(1),S(1),P(1),PFIT(1),ERR(1),AK(3)
      DIMENSION NAMEF(6)
C
C     READ OUTPUT FILE NAME.
C
10    CONTINUE
      WRITE(1,1001)
      READ(1,1002)NAMEF
C
C     SEARCH TO SEE IF FILE ALREADY EXISTS.
C
      CALL SRCH$$(K$EXST,NAMEF,12,1,0,ICODE)
      IF(ICODE .EQ. E$FNTF)GO TO 20
        WRITE(1,1006)
        READ(1,*)IOK
        IF(IOK .EQ. 1)GO TO 20
        GO TO 10
20    CONTINUE
C
C     OPEN FILE FOR WRITING.
C
      CALL SRCH$$(K$WRIT,NAMEF,12,1,0,ICODE)
      IF(ICODE .NE. 0)CALL ERRPR$(K$NRTN,ICODE,0,0,0,0)
C
C     WRITE RESULTS.
C
      WRITE(5,1003)
      WRITE(5,1004)
      WRITE(5,1005)
      DO 50 I=1,N
        WRITE(5,1010)SDB(I),S(I),P(I),PFIT(I),ERR(I)
50    CONTINUE
      WRITE(5,1007)
      WRITE(5,1008)
      WRITE(5,1009)AK(1),AK(2),AK(3)
C
C     CLOSE OUTPUT FILE.
C
      CALL SRCH$$(K$CLOS,NAMEF,12,1,0,ICODE)
      IF(ICODE .NE. 0)CALL ERRPR$(K$NRTN,ICODE,0,0,0,0)
C
      RETURN
C
1001  FORMAT('ENTER OUTPUT FILE NAME, FORMAT 6A2')
1002  FORMAT(6A2)
1003  FORMAT(///,'                    RESULTS OF QFIT')
1004  FORMAT(//,'     SNDB     ','    SN    ','      P     ','    PFIT   ',
     $'    ERR')
1005  FORMAT(/)
1006  FORMAT('QFOUT: * EXISTING FILE * IF OK TO MOD ENTER 1, ELSE 0')
1007  FORMAT(///,'                    FIT PARAMETERS           ')
1008  FORMAT(//,'                  P=1-(1+(A*SN)**B)**(-C)',//,
     $'      A       ','      B       ','      C       ')
1009  FORMAT(3(3X,E13.5))
1010  FORMAT(5(1X,F8.4,1X))
C
      END
```

```
AK       R  ARGUMENT 000011     0001S    0006S    0040
E$BCOD   I  PARAMETER           0004S
E$BDAM   I  PARAMETER           0004S
E$BFSV   I  PARAMETER           0004S
E$BFTS   I  PARAMETER           0004S
E$BKEY   I  PARAMETER           0004S
E$BNAM   I  PARAMETER           0004S
E$BOF    I  PARAMETER           0004S
E$BPAR   I  PARAMETER           0004S
E$BPAS   I  PARAMETER           0004S
E$BSGN   I  PARAMETER           0004S
E$BSUN   I  PARAMETER           0004S
E$BTRN   I  PARAMETER           0004S
E$BUFD   I  PARAMETER           0004S
E$BUNT   I  PARAMETER           0004S
E$DIRE   I  PARAMETER           0004S
E$DISK   I  PARAMETER           0004S
E$DKFL   I  PARAMETER           0004S
E$DNS    I  PARAMETER           0004S
E$DNTE   I  PARAMETER           0004S
E$DVIU   I  PARAMETER           0004S
E$EOF    I  PARAMETER           0004S
E$EXST   I  PARAMETER           0004S
E$FABT   I  PARAMETER           0004S
E$FBST   I  PARAMETER           0004S
E$FDEL   I  PARAMETER           0004S
E$FDFL   I  PARAMETER           0004S
E$FIFC   I  PARAMETER           0004S
E$FITB   I  PARAMETER           0004S
E$FIUS   I  PARAMETER           0004S
E$FNTF   I  PARAMETER           0004S    0018
E$FNTS   I  PARAMETER           0004S
E$FONC   I  PARAMETER           0004S
E$FUIU   I  PARAMETER           0004S
E$IREM   I  PARAMETER           0004S
E$NASS   I  PARAMETER           0004S
E$NATT   I  PARAMETER           0004S
E$NMLG   I  PARAMETER           0004S
E$NPHA   I  PARAMETER           0004S
E$NRIT   I  PARAMETER           0004S
E$NTIM   I  PARAMETER           0004S
E$NTSD   I  PARAMETER           0004S
E$NTUD   I  PARAMETER           0004S
E$NULL   I  PARAMETER           0004S
E$OLDP   I  PARAMETER           0004S
E$PTRM   I  PARAMETER           0004S
E$RLDN   I  PARAMETER           0004S
E$ROOM   I  PARAMETER           0004S
E$SDER   I  PARAMETER           0004S
E$SEMO   I  PARAMETER           0004S
E$SHUT   I  PARAMETER           0004S
E$SUNO   I  PARAMETER           0004S
E$TMRU   I  PARAMETER           0004S
E$TMUL   I  PARAMETER           0004S
E$UIUS   I  PARAMETER           0004S
E$UNOP   I  PARAMETER           0004S
ERR      R  ARGUMENT  000007    0001S    0006S    0036
ERRPR$   R  EXTERNAL  000000    0028     0045
I        I            000622    0035M    0036
ICODE    I            000623    0017A    0018     0027A    0028A    0044A    0045A
IOK      I            000624    0020M    0021
K$ALLD   I  PARAMETER           0003S
K$CACC   I  PARAMETER           0003S
K$CLOS   I  PARAMETER           0003S    0044
K$CONV   I  PARAMETER           0003S
K$CURR   I  PARAMETER           0003S
K$DELE   I  PARAMETER           0003S
K$DMPB   I  PARAMETER           0003S
K$DTIM   I  PARAMETER           0003S
K$ENTR   I            000000    0003S
K$EXST   I  PARAMETER           0003S    0017
K$FREE   I  PARAMETER           0003S
```

```
K$FULL  I  PARAMETER          0003S
K$GOND  I  PARAMETER          0003S
K$GPOS  I  PARAMETER          0003S
K$HOME  I  PARAMETER          0003S
K$ICUR  I  PARAMETER          0003S
K$IMFD  I  PARAMETER          0003S
K$IRTN  I  PARAMETER          0003S
K$ISEG  I  PARAMETER          0003S
K$IUFD  I  PARAMETER          0003S
K$MENT  I            000000   0003S
K$MSIZ  I  PARAMETER          0003S
K$MVNT  I  PARAMETER          0003S
K$NAME  I  PARAMETER          0003S
K$NDAM  I  PARAMETER          0003S
K$NRTN  I  PARAMETER          0003S   0028    0045
K$NSAM  I  PARAMETER          0003S
K$NSGD  I  PARAMETER          0003S
K$NSGS  I  PARAMETER          0003S
K$POSA  I  PARAMETER          0003S
K$POSN  I  PARAMETER          0003S
K$POSR  I  PARAMETER          0003S
K$PREA  I  PARAMETER          0003S
K$PRER  I  PARAMETER          0003S
K$PROT  I  PARAMETER          0003S
K$RDWR  I  PARAMETER          0003S
K$READ  I  PARAMETER          0003S
K$RPOS  I  PARAMETER          0003S
K$RSUB  I  PARAMETER          0003S
K$RWLK  I  PARAMETER          0003S
K$SENT  I            000000   0003S
K$SETC  I  PARAMETER          0003S
K$SETH  I  PARAMETER          0003S
K$SPOS  I  PARAMETER          0003S
K$SRTN  I  PARAMETER          0003S
K$TRNC  I  PARAMETER          0003S
K$UPOS  I  PARAMETER          0003S
K$WRIT  I  PARAMETER          0003S   0027
N       I  ARGUMENT  000010   0001S   0035
NAMEF   I            000013   0007S   0013M   0017A   0027A   0044A
P       R  ARGUMENT  000005   0001S   0006S   0036
PFIT    R  ARGUMENT  000006   0001S   0006S   0036
QFOUT   R            000000   0001S
S       R  ARGUMENT  000004   0001S   0006S   0036
SDB     R  ARGUMENT  000003   0001S   0006S   0036
SRCH$$  R  EXTERNAL  000000   0017    0027    0044

_10                  000021   0011D   0022
_1001                000274   0012    0049D
_1002                000320   0013    0050D
_1003                000324   0032    0051D
_1004                000351   0033    0052D
_1005                000413   0034    0054D
_1006                000416   0019    0055D
_1007                000454   0038    0056D
_1008                000504   0039    0057D
_1009                000572   0040    0059D
_1010                000602   0036    0060D
_20                  000070   0018    0021    0023D
_50                  000205   0035    0037D

0000 ERRORS [<QFOUT >FTN-REV15.3]
```

```
      SUBROUTINE QPDFIT (PD,SDB,N,P)
      DIMENSION S(15),Y(2),Z(2),W(15),PD(1),P(3),SDB(15)
      DIMENSION PDD(200)
      REAL MY,MZ
      DO 100 I=1,N
           S(I)=(SDB(I))
           PDD(I)=PD(I)/100.0
  100 CONTINUE
C
C     COMPUTE SLOPE BETWEEN FIRST TWO POINTS
C
      DO 150 I=1,2
           Y(I)=ALOG10(-ALOG10(1-PDD(I)))
  150 CONTINUE
      MY=(Y(2)-Y(1))/(S(2)-S(1))
C
C     COMPUTE SLOPE BETWEEN LAST TWO POINTS
C
      NM1=N-1
      DO 200 I=NM1,N
           Z(I)=-ALOG10(1-PDD(I))
  200 CONTINUE
      MZ=(Z(N)-Z(NM1))/(S(N)-S(NM1))
C
C     COMPUTE INITIAL ESTIMATE OF K3
C
      P(3)=MZ/MY
C
C     DISTORT THE PD VALUES
C
      DO 225 I=1,N
           PDD(I)=1-((1-PDD(I))**(1/P(3)))
  225 CONTINUE
C
C     COMPUTE LARGE PD SLOPE RATIO
C
      SRL1=ALOG10(((1-PDD(N))**(-1/P(3)))-1)
      SRL2=ALOG10(((1-PDD(NM1))**(-1/P(3)))-1)
      SRL3=(-1/P(3))*ALOG10(1-PDD(N))
      SRL4=(1/P(3))*ALOG10(1-PDD(NM1))
      SRL=(SRL1-SRL2)/(SRL3+SRL4)
C
C     COMPUTE THE SMALL PD SLOPE RATIO
C
      SRS1=ALOG10(((1-PDD(2 )**(-1/P(3)))-1)
      SRS2=ALOG10(((1-PDD(1 )**(-1/P(3)))-1)
      SRS3=ALOG10((-ALOG(10.0)/P(3))*(ALOG10(1-PDD(2))))
      SRS4=ALOG10((-ALOG(10.0)/P(3))*(ALOG10(1-PDD(1))))
      SRS=(SRS1-SRS2)/(SRS3-SRS4)
      P(3)=P(3)*(SRL/SRS)
C
C     INITIALIZE COUNTERS FOR LINEAR REGRESSION
C
      SUM=0
      SUMX=0.0
      SUMY=0.0
      SUMX2=0.0
      SUMXY=0.0
C
C     COMPUTE W(I) AND PARAMETERS FOR A STRAIGHT LINE FIT
C
      DO 250 I=1,N
           W(I)=ALOG10(((1-.01*PD(I))**(-1/P(3)))-1)
           SUM=SUM+1.0
           SUMX=SUMX+S(I)
           SUMY=SUMY+W(I)
           SUMX2=SUMX2+S(I)*S(I)
           SUMXY=SUMXY+S(I)*W(I)
  250 CONTINUE
C
      D=SUM*SUMX2-SUMX*SUMX
      B=(SUMX2*SUMY-SUMX*SUMXY)/D
```

```
      A=(SUMXY*SUM-SUMX*SUMY)/D
C
C     COMPUTE K1 AND K2
C
      P(1)=10.**(B/(10.*A))
      P(2)=10.*A
C
      WRITE (1,400) P(1),P(2),P(3)
  400 FORMAT (3HK1=,F6.4,2X,3HK2=,F6.4,2X,3HK3=,F6.4)
      RETURN
      END
```

```
A       R           001762    0073M   0077    0078
ALOG    R EXTERNAL  000000    0047    0048
ALOG10  R EXTERNAL  000000    0013    0021    0037    0038    0039    0040    0045
                              0046    0047    0048    0063
B       R           001764    0072M   0077
D       R           001766    0071M   0072    0073
I       I           001770    0005M   0006    0007    0012M   0013    0020M   0021
                              0031M   0032    0062M   0063    0065    0066    0067
                              0068
MY      R           001773    0004S   0015M   0027
MZ      R           001775    0004S   0023M   0027
N       I ARGUMENT  000005    0001S   0005    0019    0020    0023    0031    0037
                              0039    0062
NM1     I           001777    0019M   0020    0023    0038    0040
P       R ARGUMENT  000006    0001S   0002S   0027M   0032    0037    0038    0039
                              0040    0045    0046    0047    0048    0050M   0063
                              0077M   0078M   0080
PD      R ARGUMENT  000003    0001S   0002S   0007    0063
PDD     R           000010    0003S   0007M   0013    0021    0032M   0037    0038
                              0039    0040    0045    0046    0047    0048
QPDFIT  R           000000    0001S
S       R           000630    0002S   0006M   0015    0023    0065    0067    0068
SDB     R ARGUMENT  000004    0001S   0002S   0006
SRL     R           002006    0041M   0050
SRL1    R           002010    0037M   0041
SRL2    R           002012    0038M   0041
SRL3    R           002014    0039M   0041
SRL4    R           002016    0040M   0041
SRS     R           002020    0049M   0050
SRS1    R           002022    0045M   0049
SRS2    R           002024    0046M   0049
SRS3    R           002026    0047M   0049
SRS4    R           002030    0048M   0049
SUM     R           002032    0054M   0064M   0071    0073
SUMX    R           002034    0055M   0065M   0071    0072    0073
SUMX2   R           002036    0057M   0067M   0071    0072
SUMXY   R           002040    0058M   0068M   0072    0073
SUMY    R           002042    0056M   0066M   0072    0073
W       R           000666    0002S   0063M   0066    0068
Y       R           000724    0002S   0013M   0015
Z       R           000730    0002S   0021M   0023

_100                001761    0005    0008D
_150                001015    0012    0014D
_200                001065    0020    0022D
_225                001165    0031    0033D
_250                001572    0062    0069D
_400                001731    0080    0081D

0000 ERRORS [<QPDFIT>FTN-REV15.3]
```

```
      SUBROUTINE VBAR
C
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      DIMENSION SNR(12),PDF(12),TH(3),DEL(3),DINC(3),W(3,12)
C
      COMMON/COM/SNR,PDF,TH,DEL,DINC,W,M,II,JV
C
C
      DO 10 I=1,JV
      PDF(I)=1.0+((1.DO/TH(1))*SNR(I))**TH(2)
      IF(PDF(I).LT.1.DO)PDF(I)=1.DO
      PDF(I)=1.0/PDF(I)**TH(3)
      PDF(I)=1.-PDF(I)
   10 CONTINUE
C
      RETURN
      END
```

DEL	D	/COM/	000154	0006S	0008S				
DINC	D	/COM/	000170	0006S	0008S				
I	I		000070	0011M	0012	0013	0014	0015	
II	I	/COM/	000425	0008S					
JV	I	/COM/	000426	0008S	0011				
M	I	/COM/	000424	0008S					
PDF	D	/COM/	000060	0006S	0008S	0012M	0013M	0014M	0015M
SNR	D	/COM/	000000	0006S	0008S	0012			
TH	D	/COM/	000140	0006S	0008S	0012	0014		
VBAR	D		000000	0001S					
W	D	/COM/	000204	0006S	0008S				
_10			000055	0011	0016D				

0000 ERRORS [<VBAR >FTN-REV15.3]

APPENDIX F

Quick PDFIT

by J. N. Bucknam

TSC Memorandum TSC-W7-73

Technology Service Corporation

TO:	Distribution	DATE:	29 October 1979
FROM:	J. N. Bucknam	REF:	TSC-W7-73/rad
SUBJECT:	Quick PDFIT	Dist:	W. Rivers R. Blase D. Brandt J. Bucknam

1.0 INTRODUCTION AND SUMMARY

Here is a "quick and dirty" algorithm for finding approximate values for the parameters K_1, K_2, and K_3 in a Khoury-function [1] of the form

$$P_D = 1 - [1 + (K_1 S)^{K_2}]^{-K_3}. \tag{1}$$

where P_D = detection probability expressed as a fraction

S = ratio of signal power to background power.

Given a vector of pairs (P_D, S), the algorithm computes K_3 using the two largest and the two smallest values of P_D. The parameters K_1, and K_2 consistent with this K_3 are then found from the full set of data pairs.

The algorithm might be useful for computing starting values in program PDFIT [1].

The parameter estimates also are useful "as is" for approximate calculations. For the very few cases tried, the values of K_1, K_2, and K_3 obtained resulted in less than about 0.2 dB error in the required signal to noise ratio obtained from the inverse of equation (1).

2.0 ALGORITHM DERIVATION

The Khoury-function of equation (1) is easily transformed into an equivalent logarithmic Khoury-function (also known as a Bucknam-function) given by

$$\log_{10}[(1-P_D)^{-1/K_3} - 1] = K_1' + K_2' \text{ SDB} \qquad (2)$$

where

$$K_2' = K_2/10 \qquad (3)$$

$$K_1' = K_2 \log_{10} k_1 \qquad (4)$$

$$\text{SDB} = 10 \log_{10} S \qquad (5)$$

The asymptotic behavior of the left-hand side of equation (2) is

$$\log_{10}[(1-P_D)^{-1/K_3} - 1] \approx \begin{cases} -\frac{1}{K_3} \log_{10}(1-P_D), \quad P_D \to 1.0 & (6a) \\ \log_{10}[-\frac{\ln 10}{K_3} \log_{10}(1-P_D)], \quad P_D \to 0.0 & (6b) \end{cases}$$

This is illustrated in Figure 1. Thus, if we were to plot the quantity $-\log_{10}(1-P_D)$ vs. SDB, the slope of the curve would approach $K_3 \cdot K_2'$ for P_D near 1.0. Similarly, if we were to plot the quantity $\log_{10}[-\log_{10}(1-P_D)]$ vs. SDB, the slope would approach K_2' for P_D near 0. The ratio of the two slopes is an initial estimate for K_3.

This estimate of K_3 is found to underestimate the true value for K_3 greater than 1.0, and to overestimate K_3 less than 1.0. This is caused by the differences in slopes between the true function and its asymptotic forms as given in equation (6). A correction factor can be applied to the initial K_3 by computing the ratio of slopes of the asymptotes and the true function. The new value K_3' is then given by

$$K_3' = K_3 \left(\frac{SR_L}{SR_S}\right)$$

K+E SEMI-LOGARITHMIC 3 CYCLES X ... DIVISIONS KEUFFEL & ESSER CO. MADE IN U.S.A. 46 5490

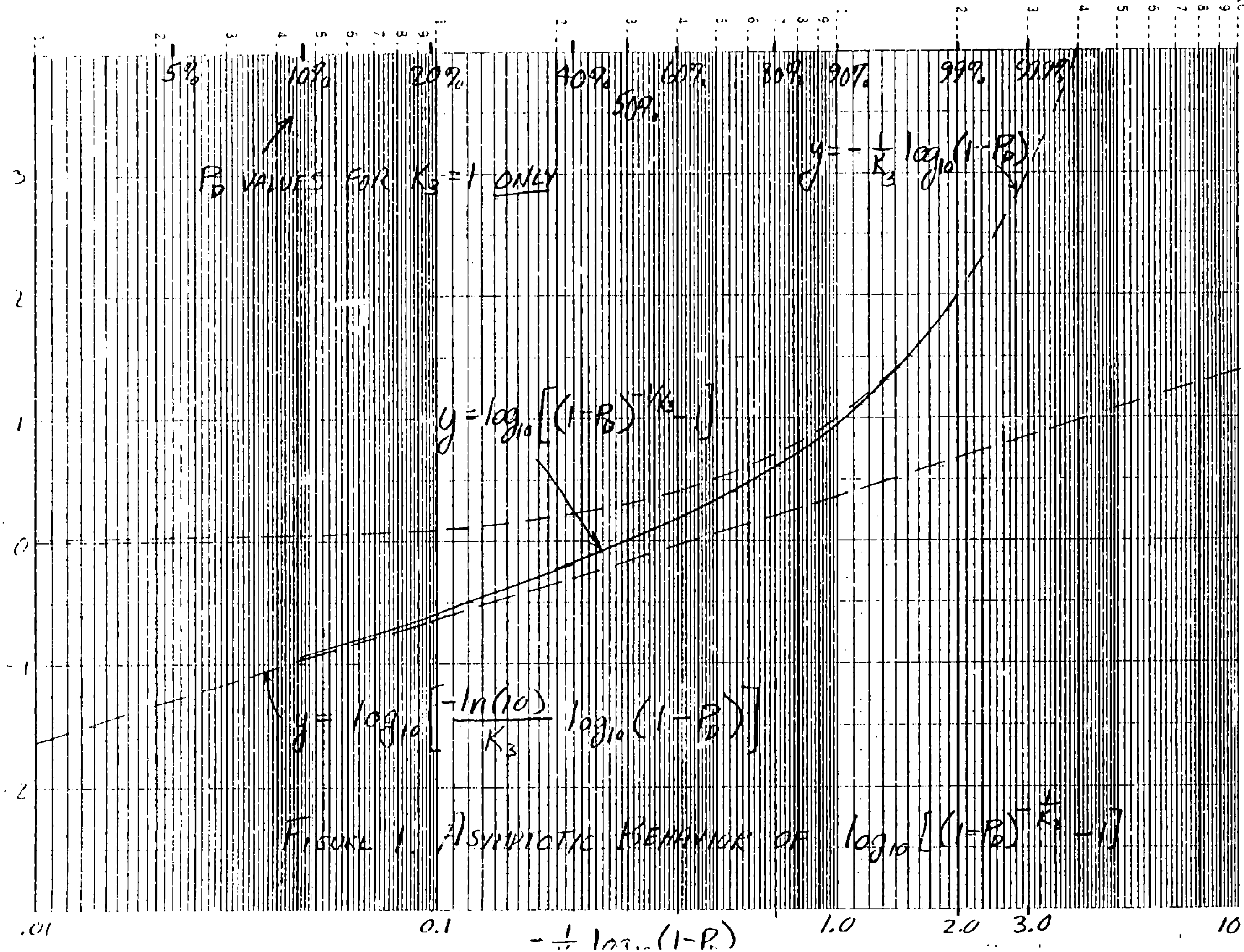

FIGURE 1. ASYMPTOTIC BEHAVIOR OF $\log_{10}\left[(1-P_b)^{-\frac{1}{K_3}} - 1\right]$

where SR_L is the ratio of the slope of the true function to the slope of the asymptotic function at large P_D, and SR_S is the ratio of slopes at small P_D.

The parameters K_1' and K_2' can then be estimated by plotting the left side of equation (2) vs. SDB using this K_3, and fitting a straight line to the data. Parameters K_1 and K_2 are then found by inversion of equations (3) and (4).

3.0 TEST CASES

The algorithm has been tried on four test cases. The test cases are the same four as in ref [2]. The least-squares fit parameter values and the "quick-and-dirty" estimates are compared in Table 1. The errors incurred in predicting required signal-to-noise ratio from the inverse function using the quick-and-dirty parameters are summarized in Tables 2-5. Note that although there are significant differences between the least-squares and approximate parameter values, the difference in required S/N is always less than about 0.2 dB in the four test cases.

```
        PDFIT
INPUT S/N VECTOR IN DB
⎕:
        SNR1
INPUT PD VECTOR IN PERCENT
⎕:
        PD1
K1,K2,K3
0.3389 2.181 0.3863
```

PD(PCT)	SNR(DB)	SNFIT	ERROR(DB)
5.053	0.622	0.8353	0.2133
10.84	2.48	2.585	0.1046
18.95	4.024	4.052	0.02799
29.01	5.438	5.408	¯0.03042
39.64	6.742	6.672	¯0.06981
50.46	8.06	7.966	¯0.09388
61.1	9.49	9.384	¯0.1059
69.97	10.91	10.81	¯0.1048
80.17	13.09	13.01	¯0.08689
89.95	16.57	16.54	¯0.03299
94.76	19.88	19.9	0.0197
98.77	27.21	27.37	0.1591

Table 2 - Swerling 1, 4 hits, $P_{FA} = 10^{-6}$

```
        PDFIT
INPUT S/N VECTOR IN DB
⎕:
        SNR2
INPUT PD VECTOR IN PERCENT
⎕:
        PD2
K1,K2,K3
0.7032 2.248 0.3738
```

PD(PCT)	SNR(DB)	SNFIT	ERROR(DB)
5.46	¯2.203	¯1.987	0.2164
10.37	¯0.675	¯0.5535	0.1215
19.48	1.035	1.062	0.02743
29.4	2.393	2.361	¯0.03215
39.86	3.657	3.585	¯0.0724
50.47	4.939	4.841	¯0.09843
60.93	6.334	6.224	¯0.1104
69.65	7.72	7.611	¯0.1092
79.77	9.852	9.761	¯0.09064
89.59	13.26	13.22	¯0.03929
94.51	16.51	16.53	0.01833
98.69	23.77	23.93	0.1688

Table 3 - Swerling 1, 11 hits, $P_{FA} = 10^{-6}$

```
      PDFIT
INPUT S/N VECTOR IN DB
⎕:
      SNR3
INPUT PD VECTOR IN PERCENT
⎕:
      PD3
K1,K2,K3
0.2294 2.521 8.309
```

PD(PCT)	SNR(DB)	SNFIT	ERROR(DB)
5.012	¯2.325	¯2.361	¯0.03551
10.75	¯0.972	¯0.9867	¯0.01471
20.93	0.273	0.2749	0.00185
29.55	0.966	0.9756	0.009561
39.37	1.59	1.606	0.01563
49.37	2.135	2.154	0.01946
59.7	2.655	2.677	0.02153
70.51	3.197	3.218	0.02137
79.51	3.688	3.707	0.01871
89.72	4.395	4.403	0.008202
94.99	4.96	4.955	¯0.00544
99.44	6.207	6.146	¯0.06067

Table 4 - Swerling 0, 7 hits, $P_{FA} = 10^{-3}$

```
      PDFIT
INPUT S/N VECTOR IN DB
⎕:
      SNR4
INPUT PD VECTOR IN PERCENT
⎕:
      PD4
K1,K2,K3
0.1306 3.035 0.7799
```

PD(PCT)	SNR(DB)	SNFIT	ERROR(DB)
5.523	3.72	3.736	0.01591
10.87	4.793	4.8	0.007345
19.01	5.756	5.758	0.001807
30.23	6.672	6.668	¯0.003626
38.56	7.235	7.228	¯0.006625
49.07	7.896	7.888	¯0.008104
59.77	8.579	8.569	¯0.009751
70.41	9.338	9.329	¯0.008807
80.67	10.27	10.26	¯0.006994
89.71	11.53	11.52	¯0.001345
94.38	12.67	12.68	0.004006
97.34	14.44	14.46	0.01619

Table 5 - Swerling 2, 4 hits, $P_{FA} = 10^{-8}$

Table 1 - Least Squares and Approximate Parameter Values for Four Test Cases

Swerling Case	# of hits	P_{FA}	$\hat{K}_1$	K_1	$\hat{K}_2$	K_2	$\hat{K}_3$	K_3	Maximum Error (dB)
1	4	10^{-6}	.3389	.3198	2.181	2.038	.3863	.4210	.21
1	11	10^{-6}	.7032	.6624	2.248	2.088	.3738	.4101	.21
0	7	10^{-3}	.2294	.2583	2.521	2.563	8.309	6.533	.06
2	-4	10^{-8}	.1806	.1790	3.035	3.010	.7799	.7925	.02

$\hat{K}_i$ = approximate parameter value

K_i = least squares parameter value

4.0 ALGORITHM COOKBOOK DESCRIPTION

Step 1. For the two smallest values of P_D(P_1 and P_2, $P_1 < P_2$), plot y_i vs SDB_i, where

$$y_i \triangleq \log_{10}[-\log_{10}(1 - P_i)] \tag{7}$$

$$SDB_i \triangleq 10 \log_{10}(S_i) \tag{8}$$

Determine the slope of the line connecting these two points, i.e.,

$$m_y = \frac{y_2 - y_1}{SDB_2 - SDB_1} \tag{9}$$

Step 2. For the two largest values of P_D(P_3 and P_4, $P_3 < P_4$), plot z_i vs. SDB_i, where

$$z_i \triangleq -\log_{10}(1 - P_i) \tag{10}$$

Determine the slope of the line connecting these two points, i.e.

$$m_z = \frac{z_4 - z_3}{SDB_4 - SDB_3} \tag{11}$$

Step 3. The initial estimate for K_3 is found as

$$K_3 = \frac{m_z}{m_y} \tag{12}$$

Step 4. Compute the correction factor.

4a. Distort the P_D values:

$$\tilde{P}_i = 1 - (1 - P_i)^{1/K_3}, \quad i = 1, 2, 3, 4$$

4b. Compute large P_D slope ratio:

$$SRL = \frac{\log_{10}[(1-\tilde{P}_4)^{-1/K_3} - 1] - \log_{10}[(1-\tilde{P}_3)^{-1/K_3} - 1]}{-1/K_3 \log_{10}[1-\tilde{P}_4] + 1/K_3 \log_{10}[1-P_3]}$$

4c. Compute the small P_D slope ratio:

$$\overset{S}{SRL} = \frac{\log_{10}[(1-\tilde{P}_2)^{-1/K_3} - 1] - \log_{10}[(1-\tilde{P}_1)^{-1/K_3} - 1]}{\log_{10}\left[\frac{-(\ln 10)}{K_3}\log_{10}(1-\tilde{P}_2)\right] - \log_{10}\left[-\left(\frac{\ln 10}{K_3}\right)\log_{10}(1-\tilde{P}_1)\right]}$$

4d. Compute the correction factor and apply it to K_3:

$$K_3' = K_3 \frac{(SRL)}{(SRS)}$$

Step 5. For all values of P_D, plot w_i vs. SDB_i, where

$$w_i \triangleq \log_{10}[(1-P_i)^{-1/K_3'} - 1] \tag{13}$$

Fit a straight line through these points, computing the slope and intercept

$$\text{slope} = a = \frac{\Delta W}{\Delta SDB} \tag{14}$$

$$\text{intercept} = b = w(SDB = 0) \tag{15}$$

Step 6. The estimates for K_1 and K_2 are found from the slope and intercept values:

$$K_2 = 10a \tag{16}$$

$$K_1 = 10^{\left(\frac{b}{10a}\right)} \tag{17}$$

An APL program implementation is listed in Figure 2.

```
     ∇PDFIT[⎕]∇
   ∇ PDFIT
[1]    'INPUT S/N VECTOR IN DB' ⋄ SNR←⎕
[2]    'INPUT PD VECTOR IN PERCENT' ⋄ PDPCT←⎕
[3]    PD←PDPCT×0.01
[4]    MT←(-/10⍟-10⍟1-PD[1 2])÷-/SNR[1 2]
[5]    MZ←(-/-10⍟1-¯2↑PD)÷-/¯2↑SNR
[6]    K3←MZ÷MT
[7]    K3←K3×K3 FACTOR PD
[8]    Y←10⍟((1-PD)*-÷K3)-1
[9]    X←SNR
[10]   K←X LINREG Y
[11]   K2←10×K[1]
[12]   K1←10*K[2]÷K2
[13]   'K1,K2,K3' ⋄ ⎕←K1,K2,K3
[14]   SNRQ←SNFIT PDPCT
[15]   'PD(PCT)       SNR(DB)       SNFIT       ERROR(DB)'
[16]   ((PDPCT,[1.5] SNR),[2] SNRQ),[2] SNRQ-SNR
   ∇


     ∇FACTOR[⎕]∇
   ∇ Z←K3 FACTOR PD
[1]   ⍝DISTORT PD BY K3
[2]    PDL←1-(1-¯2↑PD)*÷K3
[3]    PDS←1-(1-2↑PD)*÷K3
[4]   ⍝COMPUTE LARGE PD SLOPE RATIO
[5]    Y←-/10⍟((1-PDL)*-÷K3)-1
[6]    SRL←Y÷-/-(÷K3)×10⍟1-PDL
[7]   ⍝COMPUTE SMALL PD SLOPE RATIO
[8]    Y←-/10⍟((1-PDS)*-÷K3)-1
[9]    SRS←Y÷-/10⍟-((⍟10)÷K3)×10⍟1-PDS
[10]   Z←SRL÷SRS
   ∇


     ∇SNFIT[⎕]∇
   ∇ Z←SNFIT PD
[1]    PD←PD×0.01
[2]    Z←10×10⍟(÷K1)×(((1-PD)*-÷K3)-1)*÷K2
   ∇
     ∇LINREG[⎕]∇
   ∇ K←X LINREG Y
[1]    SX←+/X ⋄ SXX←+/X×X ⋄ SXY←+/X×Y ⋄ SY←+/Y ⋄ N←⍴X
[2]    A← 2 2 ⍴SX,N,SXX,SX
[3]    V←SY,SXY
[4]    K←(⌹A)+.×V
   ∇
```

Figure 2. APL Program Documentation

5.0 REFERENCES

[1] E.N. Khoury, "An Overview of Program PDFIT", TSC-W21-40, 13 June 1979.

[2] D. M. Brandt, "Radar Integration Efficiency", TSC-W53-01, 9 October 1979.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Review of Representation Functions for Probability of Detection.		5. TYPE OF REPORT & PERIOD COVERED Technical rept.
		6. PERFORMING ORG. REPORT NUMBER TSC-W25-160
7. AUTHOR(s) Wayne Rivers, J. N. Bucknam, E. N. Khoury and R. E. Blase		8. CONTRACT OR GRANT NUMBER(s) N60921-79-C-A306
9. PERFORMING ORGANIZATION NAME AND ADDRESS Technology Service Corporation 8555 16th Street, Suite 300 Silver Spring, Maryland 20910		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N00014-78-C-7065
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 11 January 1980
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 117		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ABSTRACT ON REVERSE SIDE

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

390721

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20.

A REVIEW OF REPRESENTATION FUNCTIONS
FOR PROBABILITY OF DETECTION

by

Wayne Rivers, J. N. Bucknam,
E. N. Khoury, and R. E. Blase

ABSTRACT

Simple representation functions that interrelate the primary signal detection variables for the receiver structure of Marcum and Swerling are reviewed in regard to accuracy, complexity and inversion. Functions reviewed include those of Brooks, Neuvy, and Khoury-Bucknam. The first two of these are simple algorithms for computing minimum ratio of signal energy to noise power density as functions of number of samples integrated, the target distribution case, and required probabilities of detection and false alarm. The Khoury-Bucknam functions are analytic and invertable, and they relate probability of detection and signal-to-noise ratio using parameters chosen uniquely for each case, number of samples, and probability of false alarm.

The procedures and software that support determination of the coefficients of the Khoury-Bucknam function are documented.